

Faster Lossy Generalized Flow via Interior Point Algorithms*

Samuel I. Daitch
Department of Computer Science
Yale University

Daniel A. Spielman
Program in Applied Mathematics and
Department of Computer Science
Yale University

April 7, 2008

Abstract

We present asymptotically faster approximation algorithms for the generalized flow problems in which multipliers on edges are at most 1. For this lossy version of the maximum generalized flow problem, we obtain an additive ϵ approximation of the maximum flow in time $\tilde{O}(m^{3/2} \log^2(U/\epsilon))$, where m is the number of edges in the graph, all capacities are integers in the range $\{1, \dots, U\}$, and all loss multipliers are ratios of integers in this range. For minimum cost lossy generalized flow with costs in the range $\{1, \dots, U\}$, we obtain a flow that has value within an additive ϵ of the maximum value and cost at most the optimal cost. In many parameter ranges, these algorithms improve over the previously fastest algorithms for the generalized maximum flow problem by a factor of $m^{1/2}$ and for the minimum cost generalized flow problem by a factor of approximately $m^{1/2}/\epsilon^2$.

The algorithms work by accelerating traditional interior point algorithms by quickly solving the linear equations that arise in each step. The contributions of this paper are twofold. First, we analyze the performance of interior point algorithms with approximate linear system solvers. This analysis alone provides an algorithm for the standard minimum cost flow problem that runs in time $\tilde{O}(m^{3/2} \log^2 U)$ —an improvement of approximately $\tilde{O}(n/m^{1/2})$ over previous algorithms.

Second, we examine the linear equations that arise when using an interior point algorithm to solve generalized flow problems. We observe that these belong to the family of symmetric M-matrices, and we then develop $\tilde{O}(m)$ -time algorithms for solving linear systems in these matrices. These algorithms reduce the problem of solving a linear system in a symmetric M-matrix to that of solving $\mathcal{O}(\log n)$ linear systems in symmetric diagonally-dominant matrices, which we can do in time $\tilde{O}(m)$ using the algorithm of Spielman and Teng.

All of our algorithms operate on numbers of bit length at most $\mathcal{O}(\log nU/\epsilon)$.

*This material is based upon work supported by the National Science Foundation under Grant Nos. CCF-0707522 and CCF-0634957. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

Interior-point algorithms are one of the most popular ways of solving linear programs. These algorithms are iterative, and their complexity is dominated by the cost of solving a system of linear equations at each iteration. Typical complexity analyses of interior point algorithms apply worst-case bounds on the running time of linear equations solvers. However, in most applications the linear equations that arise are quite special and may be solved by faster algorithms. Each family of optimization problem leads to a family of linear equations. For example, the maximum flow and minimum cost flow problems require the solution of linear systems whose matrices are symmetric and diagonally-dominant. The generalized versions of these flow problems result in symmetric M-matrices.

The generalized maximum flow problem is specified by a directed graph (V, E) , an inward capacity $c(e) > 0$ and a multiplier $\gamma(e) > 0$ for each edge e , and source and sink vertices s and t . For every unit flowing into edge e , $\gamma(e)$ flows out. In lossy generalized flow problems, each multiplier $\gamma(e)$ is restricted to be at most 1. In the generalized maximum flow problem, one is asked to find the flow $f : E \rightarrow \mathbb{R}^+$ that maximizes the flow into t given an unlimited supply at s , subject to the capacity constraints on the amount of flow entering each edge. In the generalized minimum cost flow problem, one also has a cost function $q(e) \geq 0$, and is asked to find the maximum flow of minimum cost (see [AMO93]).

In the following chart, we compare the complexity of our algorithms with the fastest algorithms of which we are aware. The running times are given for networks in which all capacities and costs are positive integers less than U and every loss factor is a ratio of two integers less than U . For the standard flow problems, our algorithms are exact, but for the generalized flow problems our algorithms find additive ϵ approximations, while the other approximation algorithms have multiplicative error $(1 + \epsilon)$. However, we note that our algorithms only require arithmetic with numbers of bit-length $\mathcal{O}(\log(nU/\epsilon))$, whereas we suspect that the algorithms obtaining multiplicative approximations might require much longer numbers.

In the chart, C refers to the value of the flow.

Exact algorithms	Approximation algorithms	Our algorithm
Generalized Maximum Flow $\mathcal{O}(m^2(m + n \log n) \log U)$ [GJO97] $\mathcal{O}(m^{1.5}n^2 \log(nU))$ [Vai89]	$\tilde{\mathcal{O}}(m^2/\epsilon^2)$ [FW02] $\tilde{\mathcal{O}}(m(m + n \log \log B) \log \epsilon^{-1})$ [GFNR98][TW98][FW02]	$\tilde{\mathcal{O}}(m^{1.5} \log^2(U/\epsilon))$
Generalized Minimum Cost Flow $\mathcal{O}(m^{1.5}n^2 \log(nU))$ [Vai89]	$\tilde{\mathcal{O}}(m^2 \log \log B/\epsilon^2)$ [FW02]	$\tilde{\mathcal{O}}(m^{1.5} \log^2(U/\epsilon))$
Maximum Flow $\mathcal{O}(\min(n^{3/2}, m^{1/2})m \log(n^2/m) \log U)$ [GR98]		$\tilde{\mathcal{O}}(m^{1.5} \log^2 U)$
Minimum Cost Flow $\mathcal{O}(nm \log(n^2/m) \log(nC))$ [GT87] $\mathcal{O}(nm(\log \log U) \log(nC))$ [AGOT92] $\mathcal{O}((m \log n)(m + n \log n))$ [Orl88]		$\tilde{\mathcal{O}}(m^{1.5} \log^2 U)$

1.1 The solution of systems in M-matrices

A symmetric matrix M is *diagonally dominant* if each diagonal is at least the sum of the absolute values of the other entries in its row. A symmetric matrix M is an *M-matrix* if there is a positive diagonal matrix D for which DMD is diagonally dominant. Spielman and Teng [ST04, ST06] showed

how to solve linear systems in diagonally dominant matrices to ϵ accuracy in time $\tilde{\mathcal{O}}(m \log \epsilon^{-1})$. We show how to solve linear systems in M -matrices by first computing a diagonal matrix D for which DMD is diagonally dominant, and then applying the solver of Spielman and Teng. Our algorithm for finding the matrix D applies the solver of Spielman and Teng an expected $\mathcal{O}(\log n)$ times. While iterative algorithms are known that eventually produce such a diagonal matrix D , they have no satisfactory complexity analysis [Li02, LLH⁺98, BCPT05].

1.2 Analysis of interior point methods

In our analysis of interior-point methods, we examine the complexity of the short-step dual path following algorithm of Renegar [Ren88] as analyzed by Ye [Ye97]. The key observations required by our complexity analysis are that none of the slack variables become too small during the course of the algorithm and that the algorithm still works if one $\mathcal{O}(1/\sqrt{m})$ -approximately solves each linear system in the matrix norm (defined below). Conveniently, this is the same type of approximation produced by our algorithm and that of Spielman and Teng. This is a very crude level of approximation, and it means that these algorithms can be applied very quickly. While other analyses of the behavior of interior point methods with inexact solvers have appeared [Ren96], we are unaware of any analyses that are sufficiently fine for our purposes.

This analysis is given in detail in Appendix C.

1.3 Outline of the paper

In Section 2, we describe the results of our analysis of interior point methods using approximate solvers. In Section 3, we describe the formulation of the generalized flow problems as linear programs, and discuss how to obtain the solutions from the output of an interior-point algorithm. In Section 4, we give our algorithm for solving linear systems in M -matrices.

2 Interior-Point Algorithm using an Approximate Solver

Our algorithm uses numerical methods to solve a linear program formulation of the generalized flow problems. The fastest interior-point methods for linear programs, such as that of Renegar [Ren88] require only $\mathcal{O}(\sqrt{n})$ iterations to approach the solution, where each iteration takes a step through the convex polytope by solving a system of linear equations.

In this paper, we consider stepping through the linear program using an only an approximate solver, i.e. an algorithm $\mathbf{x} = \text{Solve}(M, \mathbf{b}, \epsilon)$ that returns a solution satisfying

$$\|\mathbf{x} - M^{-1}\mathbf{b}\|_M \leq \epsilon \|M^{-1}\mathbf{b}\|_M$$

where the **matrix norm** $\|\cdot\|_M$ is given by $\|\mathbf{v}\|_M = \sqrt{\mathbf{v}^T M \mathbf{v}}$.

As mentioned above, we have analyzed the Renegar [Ren88] version of the dual path-following algorithm, along the lines of the analysis that found in [Ye97], but modified to account for the use of an approximate solver.

In particular, using the approximate solver we implement an interior-point algorithm with the following properties:

Theorem 2.1. $\mathbf{x} = \text{InteriorPoint}(A, \mathbf{b}, \mathbf{c}, \lambda_{\min}, T, \mathbf{y}^0, \epsilon)$ takes input that satisfies

- A is an $n \times m$ matrix;
- \mathbf{b} is a length n vector; \mathbf{c} is a length m vector

- AA^T is positive definite, and $\lambda_{\min} > 0$ is a lower bound on the eigenvalues of AA^T
- $T > 0$ is an upper bound on the absolute values of the coordinates in the dual linear program, i.e.

$$\|\mathbf{y}\|_\infty \leq T \text{ and } \|\mathbf{s}\|_\infty \leq T$$

$$\text{for all } (\mathbf{y}, \mathbf{s}) \text{ that satisfy } \mathbf{s} = \mathbf{c} - A^T \mathbf{y} \geq 0$$

- initial point \mathbf{y}^0 is a length n vector where $A^T \mathbf{y}^0 < \mathbf{c}$
- error parameter ϵ satisfies $0 < \epsilon < 1$

and returns $\mathbf{x} > 0$ such that $\|A\mathbf{x} - \mathbf{b}\| \leq \epsilon$ and $\mathbf{c}^T \mathbf{x} < z^* + \epsilon$.

Let us define

- U is the largest absolute value of any entry in $A, \mathbf{b}, \mathbf{c}$
- s_{\min}^0 is the smallest entry of $\mathbf{s}^0 = \mathbf{c} - A^T \mathbf{y}^0$

Then the algorithm makes $\mathcal{O}\left(\sqrt{m} \log \frac{TUm}{\lambda_{\min} s_{\min}^0 \epsilon}\right)$ calls to the approximate solver, of the form

$$\text{Solve}(AS^{-2}A^T + \mathbf{v}\mathbf{v}^T, \cdot, \epsilon')$$

where S is a positive diagonal matrix with condition number $\mathcal{O}\left(\frac{T^2 U m^2}{\epsilon}\right)$, and \mathbf{v}, ϵ' satisfy

$$\log \frac{\|\mathbf{v}\|}{\epsilon'} = \mathcal{O}\left(\log \frac{TUm}{s_{\min}^0 \epsilon}\right)$$

In Appendix C, we present a complete description of this algorithm, with analysis and proof of correctness.

3 Solving Generalized Flow

We consider network flows on a directed graph (V, E) with $V = [n]$, $E = \{e_1, \dots, e_m\}$, source $s \in V$ and sink $t \in V$. Edge e_j goes from vertex v_j to vertex w_j . and has inward capacity $c(e_j)$, flow multiplier $\gamma(e_j) < 1$, and cost $q(e_j)$.

We assume without loss of generality that t has a single in-edge, which we denote as e_t , and no out-edges.

The generalized max-flow approximation algorithm will produce a flow that sends no worse than ϵ less than the maximum possible flow to the sink.

The generalized min-cost approximation algorithm will produce a flow that, in addition to being within ϵ of a maximum flow, also has cost no greater than the minimum cost of a maximum flow (see [FW02]).

3.1 Fixing Approximate Flows

The interior-point algorithm described in the previous section produces an output that may not exactly satisfy the linear constraints $A\mathbf{x} = \mathbf{b}$. In particular, when we apply the algorithm to a network flow linear program, the output may only be an approximate flow:

Definition 3.1. *An ϵ -approximate flow approximately satisfies all capacity constraints and flow conservation constraints. In particular, every edge may have flow up to ϵ over capacity, and every vertex besides s and t may have up to ϵ excess or deficit flow.*

An exact flow satisfies all capacity constraints and has exact flow conservation at all vertices except s and t .

We are going to modify the graph slightly before running the interior-point algorithm, so that it will be easier to obtain an exact flow from the approximate flow given by the interior-point algorithm.

Let us compute the *least-lossy-paths tree* T rooted at s . This is the tree that contains, for each $v \in V - \{s, t\}$, the path $\pi_{s,v}$ from s to v that minimizes $L(v) = \prod_{e \in \pi_{s,v}} \gamma(e)^{-1}$, the factor by which the flow along the path is diminished. We can find this tree in time $\tilde{O}(m)$, using Dijkstra's algorithm to solve the single-source shortest-paths problem with edge weights $-\log \gamma(e)$.

Next, we delete from the graph all vertices v such that $L(v) > \frac{\epsilon}{2mnU}$. Note that in a maximum-flow, it is not possible to have more than $\frac{\epsilon}{2n}$ flowing into such a v , since at most mU can flow out of s . Thus, deleting each such v cannot decrease the value of the maximum flow by more than $\frac{\epsilon}{2n}$. In total, we may decrease the value of the maximum flow by at most $\frac{\epsilon}{2}$.

We define $\epsilon_{FLOW} = \frac{\epsilon^2}{64m^2n^2U^3}$. In the subsequent sections, we show how to use the interior-point method to obtain an ϵ_{FLOW} -approximate flow that has a value within ϵ_4 of the maximum flow. Assuming that the graph had been preprocessed as above, we may convert the approximate flow into an exact flow:

Lemma 3.2. *Suppose all vertices $v \in V - \{s, t\}$ satisfy $L(v) \leq \frac{\epsilon}{2mnU}$. In $\tilde{O}(m)$ time, we are able to convert an ϵ_{FLOW} -approximate flow that has a value within $\frac{\epsilon}{4}$ of the maximum flow into an exact flow that has a value within $\frac{\epsilon}{2}$ of the maximum flow. The cost of this exact flow is no greater than the cost of the approximate flow.*

Proof. Let us first fix the flows so that no vertex has more flow out than in. We use the least-lossy-paths tree T , starting at the leaves of the tree and working towards s . To balance the flow at a vertex v we increase the flow on the tree edge into v . After completing this process, for each v we will have added a path of flow that delivers at most $\frac{\epsilon^2}{64m^2n^2U^3}$ additional units of flow to v . Since $L(v) \leq \frac{\epsilon}{2mnU}$, no such path requires more than $\frac{\epsilon^2}{64m^2n^2U^3} \cdot \frac{2mnU}{\epsilon} = \frac{\epsilon}{32mnU^2}$ flow on an edge, and so in total we have added no more than $\frac{\epsilon}{32mU^2}$ to each edge.

Next, let us fix the flows so that no vertex has more flow in than out. We follow a similar procedure as above, except now we may use any spanning tree rooted at and directed towards t . Starting from the leaves, we balance the vertices by increasing flow out the tree edge. Since the network is lossy, the total amount added to each edge is at most $\frac{\epsilon^2}{64m^2n^2U^3} \cdot n \leq \frac{\epsilon^2}{64m^2nU^3}$.

Recall that we started with each edge having flow up to $\frac{\epsilon^2}{64m^2n^2U^3}$ over capacity. After balancing the flows at the vertices, each edge may now be over capacity by as much as

$$\frac{\epsilon}{32mU^2} + \frac{\epsilon^2}{64m^2nU^3} + \frac{\epsilon^2}{64m^2n^2U^3} \leq \frac{\epsilon}{16mU^2}$$

Since the edge capacities are at least 1, the flow on an edge may be as much as $(1 + \frac{\epsilon}{16mU^2})$ times the capacity.

Furthermore, while balancing the flows we may have added as much as $\frac{\epsilon}{16mU^2} \cdot mU = \frac{\epsilon^2}{16U}$ to the total cost of the flow. Assuming that the value of approximate flow was at least $\frac{\epsilon}{4}$, its cost must also have been at least $\frac{\epsilon}{4}$, and so we have increased the cost by a multiplicative factor of at most $(1 + \frac{\epsilon}{4U})$.

(If the approximate flow had value less than $\frac{\epsilon}{4}$, then the empty flow trivially solves this flow rounding problem.)

By scaling the entire flow down by a multiplicative factor of $(1 + \frac{\epsilon}{4U})^{-1}$, we solve the capacity violations, and also reduce the cost of the exact flow to be no greater than that of the approximate flow. Since the value of a flow can be at most U , the flow scaling decreases the value of the flow by no more than $\epsilon/4$, as required. \square

The above procedure produces an exact flow that is within $\epsilon/2$ of the maximum flow in the preprocessed graph, and therefore is within ϵ of the maximum flow in the original graph. Furthermore, the cost of the flow is no greater than the minimum cost of a maximum flow in the original graph.

Thus to solve a generalized flow problem, it remains for us to describe how to use the interior-point algorithm to generate a ϵ_{FLOW} -approximate flow that has a value within $\epsilon/4$ of the maximum flow, and, for the min-cost problem, also has cost no greater than the the minimum cost of a maximum flow.

3.2 Generalized Max-Flow

We formulate the maximum flow problem as a linear program as follows: Let A be the $(n-2) \times m$ matrix whose nonzero entries are $A_{v_j,j} = -1$ and $A_{w_j,j} = \gamma(e_j)$, but without rows corresponding to s and t . Let \mathbf{c} be the length m vector containing the edge capacities. Let \mathbf{u}_t be the length m unit vector with a 1 entry for edge e_t . Let the vectors \mathbf{x}_1 and \mathbf{x}_2 respectively denote the flow into each edge and the unused inward capacity of each edge. The max-flow linear program, in canonical form, is:

$$\begin{aligned} \min_{\mathbf{x}_i} \quad & -\mathbf{u}_t^T \mathbf{x}_1 \\ \text{s.t.} \quad & \begin{bmatrix} A \\ I & I \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{c} \end{bmatrix} \\ & \text{and } \mathbf{x}_i \geq 0 \end{aligned}$$

The constraint $A\mathbf{x}_1 = 0$ ensures that flow is conserved at every vertex except s and t , while the constraint $\mathbf{x}_1 + \mathbf{x}_2 = \mathbf{c}$ ensures that the capacities are obeyed.

Now, the dual of the above linear program is not bounded, which is a problem for our interior-point algorithm. To fix this, we modify the linear program slightly:

$$\begin{aligned} \min_{\mathbf{x}_i} \quad & \left(-\mathbf{u}_t^T \mathbf{x}_1 + \frac{4U}{\epsilon_{FLOW}} (\mathbf{1}_m^T \mathbf{x}_3 + \mathbf{1}_{n-2}^T \mathbf{x}_4 + \mathbf{1}_{n-2}^T \mathbf{x}_5) \right) \quad \text{s.t. } \mathbf{x}_i \geq 0 \\ & \text{and } \begin{bmatrix} A & & & I & -I \\ I & I & -I & & \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{c} \end{bmatrix} \end{aligned}$$

(We use $\mathbf{1}_k$ to denote the all-ones vector of length k .)

Lemma 3.3. *This modified linear program has the same optimum value as the original linear program.*

Proof. Let us examine the new variables in the modified program and note that \mathbf{x}_3 has the effect of modifying the capacities, while \mathbf{x}_4 and \mathbf{x}_5 create excess or deficit of flow at the vertices. Since we have a lossy network, a unit modification of any of these values cannot change the value of the flow by more than 1, and therefore must increase the value of the modified linear program. Thus, at the optimum we have $\mathbf{x}_3 = \mathbf{x}_4 = \mathbf{x}_5 = 0$ and so the solution is the same as that of the original linear program. \square

The modified linear program has the following equivalent dual linear program:

$$\begin{aligned} & \max_{\mathbf{y}_i} \mathbf{c}^T \mathbf{y}_2 \quad \text{s.t.} \quad \mathbf{s}_i \geq 0 \\ \text{and} \quad & \begin{bmatrix} A^T & I \\ & I \\ & -I \\ I & \\ -I & \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \\ \mathbf{s}_4 \\ \mathbf{s}_5 \end{bmatrix} = \begin{bmatrix} -\mathbf{u}_t \\ \mathbf{0} \\ (4U/\epsilon_{FLOW}) \cdot \mathbf{1}_m \\ (4U/\epsilon_{FLOW}) \cdot \mathbf{1}_{n-2} \\ (4U/\epsilon_{FLOW}) \cdot \mathbf{1}_{n-2} \end{bmatrix} \end{aligned}$$

Lemma 3.4. *The above dual linear program is bounded. In particular, the coordinates of all feasible dual points have absolute value at most $(nU + 1) \cdot \frac{4U}{\epsilon_{FLOW}} + 1$.*

Proof. Of the five constraints in the dual linear program, the last four give $\frac{4U}{\epsilon_{FLOW}}$ as an explicit bound on the absolute value of \mathbf{y} coordinates. It then follows that $\frac{8U}{\epsilon_{FLOW}}$ is an upper bound on the coordinates of $\mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_5$, and the coordinates of $\mathbf{s}_1 = -\mathbf{u}_t - A^T \mathbf{y}_1 - \mathbf{y}_2$ can be at most $(nU + 1) \cdot \frac{4U}{\epsilon_{FLOW}} + 1$. \square

We refer to the \mathbf{s}_i variables as the slacks. Recall that we must provide the interior-point algorithm with an initial dual feasible point \mathbf{y}^0 such that the corresponding slacks \mathbf{s}^0 are bounded away from zero. We choose the following initial point, and note that the slacks are bounded from below by $\frac{U}{\epsilon_{FLOW}}$:

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_1^0 \\ \mathbf{y}_2^0 \end{bmatrix} &= \begin{bmatrix} \mathbf{0} \\ -(2U/\epsilon_{FLOW}) \cdot \mathbf{1}_m \end{bmatrix} \\ \begin{bmatrix} \mathbf{s}_1^0 \\ \mathbf{s}_2^0 \\ \mathbf{s}_3^0 \\ \mathbf{s}_4^0 \\ \mathbf{s}_5^0 \end{bmatrix} &= \begin{bmatrix} (2U/\epsilon_{FLOW}) \cdot \mathbf{1}_m - \mathbf{u}_t \\ (2U/\epsilon_{FLOW}) \cdot \mathbf{1}_m \\ (2U/\epsilon_{FLOW}) \cdot \mathbf{1}_m \\ (4U/\epsilon_{FLOW}) \cdot \mathbf{1}_{n-2} \\ (4U/\epsilon_{FLOW}) \cdot \mathbf{1}_{n-2} \end{bmatrix} \end{aligned}$$

We must also provide the interior-point algorithm with a lower bound on the eigenvalues of the matrix

$$\begin{bmatrix} A & & & I & -I \\ & I & -I & & \\ I & I & -I & & \end{bmatrix} \begin{bmatrix} A^T & I \\ & I \\ & -I \\ I & \\ -I & \end{bmatrix} = \begin{bmatrix} AA^T + 2I & A \\ & A^T & 3I \end{bmatrix}$$

Note that we may subtract $2I$ from the above matrix and still have a positive definite matrix, so $\lambda_{\min} = 2$ is certainly a lower bound on the eigenvalues.

Using the above values for \mathbf{y}^0 and λ_{\min} , and the bound on the dual coordinates given in Lemma 3.4, we now call **InteriorPoint** on the modified max-flow linear program, using error parameter $\frac{\epsilon_{FLOW}}{2}$. In the solution returned by the interior-point algorithm, the vector \mathbf{x}_1 assigns a flow value to each edge such that the flow constraints are nearly satisfied:

Lemma 3.5. *\mathbf{x}_1 is an ϵ_{FLOW} -approximate flow with value within $\epsilon_{FLOW}/2$ of the maximum flow.*

Proof. Observe that the amount flowing into t is at least -1 times the value of the modified linear program. Since the interior-point algorithm generates a solution to the modified linear program within $\epsilon_{FLOW}/2$ of the optimum value, which is -1 times the maximum flow, the amount flowing into t surely must be within $\epsilon_{FLOW}/2$ of the maximum flow.

Now, let us note more precisely that the modified linear program aims to minimize the objective function computed by subtracting the amount flowing into t from $4U/\epsilon_{FLOW}$ times the sum of the entries of \mathbf{x}_3 , \mathbf{x}_4 , and \mathbf{x}_5 . Since the minimum value of this objective function must be negative, and the solution returned by the interior-point algorithm has a value within $\epsilon_{FLOW}/2$ of the minimum, the value of this solution must be less than $\epsilon_{FLOW}/2 < U$. The amount flowing into t is also at most U , so no entry of \mathbf{x}_3 , \mathbf{x}_4 , \mathbf{x}_5 can be greater than $2U/(4U/\epsilon_{FLOW}) = \epsilon_{FLOW}/2$.

The interior-point algorithm guarantees that

$$\|A\mathbf{x}_1 + \mathbf{x}_4 - \mathbf{x}_5\| < \frac{\epsilon_{FLOW}}{2} \quad \text{and} \quad \|\mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 - \mathbf{c}\| < \frac{\epsilon_{FLOW}}{2}$$

and so we may conclude that

$$\|A\mathbf{x}_1\| < \epsilon_{FLOW} \quad \text{and} \quad \mathbf{x}_1 \leq \mathbf{c} + \epsilon_{FLOW}$$

Indeed, this is precisely what it means for \mathbf{x}_1 to describe an ϵ_{FLOW} -approximate flow. \square

3.3 Generalized Min-Cost Flow

As a first step in solving the generalized min-cost flow problem, we solve the generalized max-flow linear program as described above, to find a value F that is within $\frac{\epsilon}{8}$ of the maximum flow.

We now formulate a linear program for finding the minimum cost flow that delivers F units of flow to t :

$$\begin{aligned} \min_{\mathbf{x}_i} \mathbf{q}^T \mathbf{x}_1 \quad & \text{s.t.} \quad \mathbf{x}_i \geq 0 \\ \text{and} \quad & \begin{bmatrix} A \\ I \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} F \cdot \mathbf{e}_t \\ \mathbf{c} \end{bmatrix} \end{aligned}$$

where \mathbf{q} is the length n vector containing the edge costs, and \mathbf{e}_t is the length $n - 1$ vector that assigns 1 to vertex t and 0 to all the other vertices except s . A is the same matrix as in the max-flow linear program, except that we include the row corresponding to t , which translates to a new constraint that F units must flow into t .

We must again modify the linear program so that the dual will be bounded:

$$\begin{aligned} \min_{\mathbf{x}_i} \left(\mathbf{q}^T \mathbf{x}_1 + \left(\frac{4mU^2}{\epsilon_{FLOW}} \right) (\mathbf{1}_m^T \mathbf{x}_3 + \mathbf{1}_{n-1}^T \mathbf{x}_4 + \mathbf{1}_{n-1}^T \mathbf{x}_5) \right) \quad & \text{s.t.} \quad \mathbf{x}_i \geq 0 \\ \text{and} \quad & \begin{bmatrix} A & & I & -I \\ I & I & -I & \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} F \cdot \mathbf{e}_t \\ \mathbf{c} \end{bmatrix} \end{aligned}$$

Lemma 3.6. *This modified linear program has the same optimum value as the original linear program.*

Proof. We examine the new variables and note that \mathbf{x}_3 modifies the capacities, while \mathbf{x}_4 and \mathbf{x}_5 create excess supply (or demand) at the vertices. A unit modification to any of these values can at best create a new path for one unit of flow to arrive at the sink. This new path has cost at least 1, and it can replace an path in the optimum flow of cost at most nU , for a net improvement in the cost of the flow of at most $nU - 1$, which is less than $\frac{4mU^2}{\epsilon_{FLOW}}$. Thus the value of the modified linear program can only increase when these new variables are set to non-zero values. \square

Now, the dual linear program is:

$$\begin{aligned} & \max_{\mathbf{y}_i} (F \cdot \mathbf{e}_t^T \mathbf{y}_1 + \mathbf{c}^T \mathbf{y}_2) \quad \text{s.t.} \quad s_i \geq 0 \\ \text{and} \quad & \begin{bmatrix} A^T & I \\ & I \\ & -I \\ I \\ -I \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \\ \mathbf{s}_4 \\ \mathbf{s}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ \mathbf{0} \\ (4mU^2/\epsilon_{FLOW}) \cdot \mathbf{1}_m \\ (4mU^2/\epsilon_{FLOW}) \cdot \mathbf{1}_{n-1} \\ (4mU^2/\epsilon_{FLOW}) \cdot \mathbf{1}_{n-1} \end{bmatrix} \end{aligned}$$

Lemma 3.7. *The above dual linear program is bounded. In particular, the coordinates of all feasible dual points have absolute value at most $(nU + 1) \cdot \frac{4mU^2}{\epsilon_{FLOW}}$.*

Proof. Of the five constraints in the dual linear program, the last four give $\frac{4mU^2}{\epsilon_{FLOW}}$ as an explicit bound on the absolute value of \mathbf{y} coordinates. It then follows that $\frac{8mU^2}{\epsilon_{FLOW}}$ is a upper bound on the coordinates of $\mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_5$, and the coordinates of $\mathbf{s}_1 = \mathbf{q} - A^T \mathbf{y}_1 - \mathbf{y}_2$ can be at most $(nU + 1) \cdot \frac{4mU^2}{\epsilon_{FLOW}}$. \square

Let us also note that $\mathbf{y}^0 = \begin{bmatrix} \mathbf{0} \\ -(mU^2/\epsilon_{FLOW}) \mathbf{1}_m \end{bmatrix}$ is an initial interior dual point with all slacks at least $\frac{mU^2}{\epsilon_{FLOW}}$.

Using the above initial point, the bound on the dual coordinates from Lemma 3.7, and $\lambda_{min} = 2$ as in the previous section, we run **InteriorPoint** on the modified min-cost linear program, with error parameter $\frac{\epsilon_{FLOW}}{2}$. In the solution returned by the interior-point algorithm, the vector \mathbf{x}_1 assigns a flow value to each edge such that the flow constraints are nearly satisfied:

Lemma 3.8. *\mathbf{x}_1 is an ϵ_{FLOW} -approximate flow with value within $\frac{5\epsilon}{32}$ of the maximum flow.*

Proof. Note that any flow in total cannot cost more than mU^2 , even if all edges are filled to maximum capacity. Therefore the value of the solution output by the interior-point algorithm can be at most $mU^2 + \frac{\epsilon_{FLOW}}{2} < 2mU^2$, and so in particular no entry of $\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5$ can be greater than $\frac{\epsilon_{FLOW}}{2}$.

Now, the interior-point algorithm guarantees that

$$\|A\mathbf{x}_1 + \mathbf{x}_4 - \mathbf{x}_5 - F \cdot \mathbf{e}_t\| < \frac{\epsilon_{FLOW}}{2} \quad \text{and} \quad \|\mathbf{x}_1 + \mathbf{x}_2 - \mathbf{x}_3 - \mathbf{c}\| < \frac{\epsilon_{FLOW}}{2}$$

and so we may conclude that

$$\|A\mathbf{x}_1 - F \cdot \mathbf{e}_t\| < \epsilon_{FLOW} \quad \text{and} \quad \mathbf{x}_1 \leq \mathbf{c} + \epsilon_{FLOW}$$

These inequalities imply that this is a ϵ_{FLOW} -approximate flow, and additionally that at least $F - \epsilon_{FLOW}$ is flowing into t . Since F is within $\frac{\epsilon}{8}$ of the maximum flow, the amount flowing into t must be within $\frac{\epsilon}{8} + \epsilon_{FLOW} < \frac{5\epsilon}{32}$ of the maximum flow. \square

By scaling down the \mathbf{x}_1 flow slightly, we obtain a flow that does not exceed the minimum cost of a maximum flow:

Lemma 3.9. $\mathbf{x}'_1 = (1 - \frac{\epsilon}{12U})\mathbf{x}_1$ is an ϵ_{FLOW} -approximate flow with value within $\frac{\epsilon}{4}$ of the maximum flow, and with cost at most the minimum cost of a maximum flow.

Proof. We may assume that the value of flow \mathbf{x}_1 is at least $\frac{3\epsilon}{32}$, because otherwise the maximum flow would have to be at most $\frac{3\epsilon}{32} + \frac{5\epsilon}{32} = \frac{\epsilon}{4}$, and so the empty flow would trivially be within $\frac{\epsilon}{4}$ of the maximum. Therefore, the minimum cost of a maximum flow must also be at least $\frac{3\epsilon}{32}$.

The interior-point algorithm guarantees that the cost of \mathbf{x}_1 does not exceed this optimum cost by more than $\frac{\epsilon_{FLOW}}{2}$, and so must also not exceed the optimum cost by a multiplicative factor of more than $(1 + \frac{16\epsilon_{FLOW}}{3\epsilon}) < (1 + \frac{\epsilon}{12U})$. Thus, $\mathbf{x}'_1 = (1 - \frac{\epsilon}{12U})\mathbf{x}_1$ must have cost below the optimum.

Furthermore, since the value of the flow \mathbf{x}_1 can be at most U , scaling down by $(1 - \frac{\epsilon}{12U})$ cannot decrease the value of the flow by more than $\frac{\epsilon}{12}$. Therefore, the value of the value \mathbf{x}'_1 is within $\frac{\epsilon}{12} + \frac{5\epsilon}{32} < \frac{\epsilon}{4}$ of the maximum. \square

3.4 Running Time

The linear systems in the above linear programs take the form

$$\bar{A} = \begin{bmatrix} A & & I & -I \\ I & I & -I & \end{bmatrix}$$

so the running time of the interior-point method depends on our ability to approximately solve systems of the form $\bar{A}S^{-2}\bar{A}^T + \mathbf{v}\mathbf{v}^T$, where diagonal matrix S and vector \mathbf{v} are as described in Theorem 2.1. As it turns out, this is not much more difficult than solving a linear system in $AS_1^{-2}A^T$, where S_1 is the upper left submatrix of S .

The matrix $AS_1^{-2}A^T$ is a symmetric M -matrix. In the next section, we describe how to approximately solve systems in such matrices in expected time $\tilde{O}(m \log \frac{\kappa}{\epsilon})$, where κ is the condition number of the matrix. We then extend this result to solve the systems $\bar{A}S^{-2}\bar{A}^T + \mathbf{v}\mathbf{v}^T$ in time $\tilde{O}(m \log \frac{\kappa\|\mathbf{v}\|}{\epsilon})$, where κ is the condition number of $\bar{A}S^{-2}\bar{A}^T$.

Theorem 3.10. Using our interior-point algorithm, we can solve the generalized max flow and generalized min-cost flow problems in time $\tilde{O}(m^{3/2} \log^2(U/\epsilon))$

Proof. According to Theorem 2.1, the interior-point algorithm requires $\mathcal{O}(\sqrt{m} \log \frac{TU_m}{\lambda_{\min} s_{\min}^0 \epsilon})$ calls to the solver.

Recall that T is an bound on the coordinates of the dual linear program, and s_{\min}^0 is the smallest slack at the initial point. Above, we gave both of these values to be polynomial in $\frac{mU}{\epsilon}$, for both the max-flow and min-cost linear programs. We also gave $\lambda_{\min} = 2$ as a lower bound on the eigenvalues of $\bar{A}\bar{A}^T$. Thus, the total number of solves is $\tilde{O}(\sqrt{m} \log \frac{U}{\epsilon})$.

Again referring to Theorem 2.1, we find that the condition number of $\bar{A}S^{-2}\bar{A}^T$ is polynomial in $\frac{mU}{\epsilon}$, as is the expression $\frac{\|\mathbf{v}\|}{\epsilon}$. We conclude that each solve takes time $\tilde{O}(m \log \frac{U}{\epsilon})$.

The preprocessing only took time $\tilde{O}(m)$ so we obtain a total running time of $\tilde{O}(m^{3/2} \log^2(U/\epsilon))$. \square

3.5 Standard Min-Cost Flow

In this section we describe how to use interior-point algorithms to give an exact solution to the standard (i.e. no multipliers on edges) min-cost flow problem.

We use the following property of the standard flow problem:

Theorem 3.11 (see [Sch03, Theorem 13.20]). *Given a flow network with integer capacities, and a positive integer F , let Ω_{FLOW} be the set of flow vectors \mathbf{x} that flow F units into t and satisfy all capacity and flow conservation constraints. Then Ω_{FLOW} is a convex polytope in which all vertices have integer coordinates.*

Our goal is to find the flow in Ω_{FLOW} of minimum cost. Since the cost function is linear, if there is a unique minimum-cost flow of value F , it must occur at a vertex of Ω_{FLOW} . By Theorem 3.11 this must be an integer flow, and we could find this flow exactly by running the interior-point algorithm until it is clear to which integer flow we are converging.

Unfortunately, the minimum-cost flow may not be unique. However, by applying the Isolation Lemma of Mulmuley, Vazirani, and Vazarani [MVV87], we can modify the cost function slightly so that the minimum-cost flow is unique, and is also a minimum-cost flow under the original cost function.

Let us first state a modified version of the Isolation Lemma:

Lemma 3.12 (see [KS01, Lemma 4]). *Given any collection of linear functions on m variables with integer coefficients in the range $\{0, \dots, U\}$. If each variable is independently set uniformly at random to a value from the set $\{0, \dots, 2mU\}$, then with probability at least $1/2$ there is a unique function in the collection that takes minimum value.*

We now describe how to force the minimum-cost flow to be unique:

Lemma 3.13. *Given a flow network with capacities and costs in the set $\{1, 2, \dots, U\}$, and a positive integer F , modify the cost of each edge independently by adding a number uniformly at random from the set $\{\frac{1}{4m^2U^2}, \frac{2}{4m^2U^2}, \dots, \frac{2mU}{4m^2U^2}\}$. Then with probability at least $1/2$, the modified network has a unique minimum-cost flow of value F , and this flow is also a minimum-cost flow of value F in the original network.*

Proof. The modified cost of a flow at a vertex of Ω_{FLOW} is a linear function of m independent variables chosen uniformly at random from the set $\{\frac{1}{4m^2U^2}, \frac{2}{4m^2U^2}, \dots, \frac{2mU}{4m^2U^2}\}$. where the coefficients are the coordinates of the flow vector, which by Lemma 3.11 are integers in the range $\{0, \dots, U\}$. So the Isolation Lemma tells us that with probability at least $1/2$, there is a unique vertex of Ω_{FLOW} with minimum modified cost.

Now, any vertex that was not originally of minimum cost must have been more expensive than the minimum cost by an integer. Since the sum of the flows on all edges can be at most mU , and no edge had its cost increased by more than $\frac{1}{2mU}$, the total cost of any flow cannot have increased by more than $1/2$. Thus, a vertex that was not originally of minimum cost cannot have minimum modified cost. \square

We may now give an exact algorithm for standard minimum-cost flow. Note that this algorithm works for any integer flow value, but in particular we may easily find the exact max-flow value by running the interior-point max-flow algorithm with an error of $1/2$, since we know the max-flow value is an integer.

Lemma 3.14. *To solve the standard minimum-cost flow problem in expected time $\tilde{O}(m^{3/2} \log^2 U)$, perturb the edge costs as in Lemma 3.13, then run the min-cost flow interior point algorithm with an error of $\frac{1}{12m^2U^3}$, and round the flow on each edge to the nearest integer.*

Proof. Let us prove correctness assuming that the modified costs do isolate a unique minimum-cost flow. The running time then follows directly from Theorem 3.10, and the fact from Lemma 3.13 that

after a constant number of tries we can expect the modified costs to yield a unique minimum-cost flow.

We first note that the modified edge costs are integer multiples of $\delta = \frac{1}{4m^2U^2}$. Therefore, by Theorem 3.11 the cost of the minimum-cost flow is at least δ less than the cost at any other vertex of Ω_{FLOW} .

Now, the flow returned by the interior-point algorithm can be expressed as a weighted average of the vertices of Ω_{FLOW} . Since the cost of this flow is within $\frac{1}{12m^2U^3} = \frac{\delta}{3U}$ of the minimum cost, this weighted average must assign a combined weight of at most $\frac{1}{3U}$ to the non-minimum-cost vertices. Therefore, the flow along any edge differs by at most $1/3$ from the minimum-cost flow. So by rounding to the nearest integer flow, we obtain the minimum-cost flow. \square

4 Solving linear systems in symmetric M-Matrices

A symmetric *M-matrix* is a positive definite symmetric matrix with non-positive off-diagonals (see, e.g. [HJ91, Axe96, BP94]). Every *M-matrix* has a factorization of the form $M = AA^T$ where each column of A has at most 2 nonzero entries [BCPT05]. Given such a factorization of an *M-matrix*, we will show how to solve linear systems in the *M-matrix* in nearly-linear time. Throughout this section, M will be an $n \times n$ symmetric *M-matrix* and A will be a $n \times m$ matrix with 2 nonzero entries per column such that $M = AA^T$. Note that M has $\mathcal{O}(m)$ non-zero entries.

Our algorithm will make use of the Spielman-Teng $\tilde{\mathcal{O}}(m)$ expected time approximate solver for linear systems in symmetric diagonally-dominant matrices, where we recall that a symmetric matrix is *diagonally-dominant* if each diagonal is at least the sum of the absolute values of the other entries in its row. It is *strictly diagonally-dominant* if each diagonal exceeds each corresponding sum.

We will use the following standard facts about symmetric *M-matrices*, which can be found, for example, in [HJ91]:

Fact 4.1. If $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix}$ is a symmetric *M-matrix* with M_{11} a principal minor, then:

1. M is invertible and M^{-1} is a nonnegative matrix.
2. M_{12} is a nonpositive matrix.
3. M_{11} is an *M-matrix*.
4. The **Schur complement** $S = M_{22} - M_{12}^T M_{11}^{-1} M_{12}$ is an *M-matrix*.
5. If all eigenvalues of M fall in the range $[\lambda_{\min}, \lambda_{\max}]$, then so do all diagonal entries of S .
6. For any positive diagonal matrix D , DMD is an *M-matrix*.
7. There exists a positive diagonal matrix D such that DMD is strictly diagonally-dominant.

Our algorithm will work by finding a diagonal matrix D for which DMD is diagonally-dominant, providing us with a system to which we may apply the solver of Spielman and Teng. Our algorithm builds D by an iterative process. In each iteration, it decreases the number of rows that are not dominated by their diagonals by an expected constant factor. The main step of each iteration involves the solution of $\mathcal{O}(\log n)$ diagonally-dominant linear systems. For simplicity, we first explain how our algorithm would work if we made use of an algorithm $\mathbf{x} = \text{ExactSolve}(M, \mathbf{b})$ that exactly solves the system $M\mathbf{x} = \mathbf{b}$, for diagonally-dominant M . We then explain how we may substitute an approximate solver.

The key to our analysis is the following lemma, which says that if we multiply an *M-matrix* by a random diagonal matrix, then a constant fraction of the diagonals probably dominate their rows.

Lemma 4.2 (Random Scaling Lemma). *Given an $n \times n$ M -matrix M , and positive real values $\zeta \leq 1$ and $r \leq \frac{1}{4}$, let D be a random diagonal $n \times n$ matrix where each diagonal entry d_i is chosen independently and uniformly from the interval $(0, 1)$.*

Let $T \subset [n]$ be the set of rows of MD with sums at least r times the pre-scaled diagonal, i.e.

$$T = \{i \in [n] : (MD\mathbf{1})_i \geq rm_{ii}\}$$

With probability at least $\frac{1-4r}{4r+7}$, we have

$$|T| \geq \left(\frac{1}{8} - \frac{r}{2}\right) \left(1 - \beta - \frac{2}{3\zeta}\right) n$$

where β is the fraction of the diagonal entries of M that are less than ζ times the average diagonal entry.

Note in particular that for $r = 0$, T is the set of rows dominated by their diagonals.

We will use the Random Scaling Lemma to decrease the number of rows that are not dominated by their diagonals. We will do this by preserving the rows that are dominated by their diagonals, and applying this lemma to the rest. Without loss of generality we write $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix} = \begin{bmatrix} A_1 A_1^T & A_1 A_2^T \\ A_2 A_1^T & A_2 A_2^T \end{bmatrix}$, where the rows in the top section of M are the ones that are already diagonally-dominant, so in particular M_{11} is diagonally-dominant. Let $S = M_{22} - M_{12}^T M_{11}^{-1} M_{12}$ be the Schur complement and let S_D be the matrix containing only the diagonal entries of S .

We construct a random diagonal matrix D_R of the same size as M_{22} by choosing each diagonal element independently and uniformly from $(0, 1)$. We then create diagonal matrix $D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}$ where $D_2 = S_D^{-1/2} D_R$ and the diagonal entries of D_1 are given by $-M_{11}^{-1} M_{12} D_2 \mathbf{1}$. We know that the diagonal entries of D_1 are positive because Fact 4.1 tells us that M_{11}^{-1} is nonnegative and M_{12} is nonpositive.

We now show that the first set of rows of DMD are diagonally-dominant, and a constant fraction of the rest probably become so as well. Since M is an M -matrix and D is positive diagonal, DMD has no positive off-diagonals. Therefore, the diagonally-dominant rows of DMD are the rows with nonnegative row sums. The row sums of DMD are:

$$\begin{aligned} DMD\mathbf{1} &= \begin{bmatrix} D_1 M_{11} D_1 \mathbf{1} + D_1 M_{12} D_2 \mathbf{1} \\ D_2 M_{12}^T D_1 \mathbf{1} + D_2 M_{22} D_2 \mathbf{1} \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ D_2 S D_2 \mathbf{1} \end{bmatrix} = \begin{bmatrix} 0 \\ D_R S_D^{-1/2} S S_D^{-1/2} D_R \mathbf{1} \end{bmatrix} \end{aligned}$$

Note that the diagonal entries of $S_D^{-1/2} S S_D^{-1/2}$ are all 1. Thus by invoking Lemma 4.2 with $r = 0$ and $\zeta = 1$, we find that there is a $1/7$ probability that at least $1/24$ of the row sums in the bottom section of DMD become nonnegative. Furthermore, we see that row sums in the top section remain nonnegative.

The only problem with this idea is that in each iteration it could take $\tilde{O}(mn)$ time to compute the entire matrix S . Fortunately, we actually only need to compute the diagonals of S , (i.e. the matrix S_D). In fact, we only actually need a diagonal matrix Σ that approximates S_D . As long as the diagonals of $\Sigma^{-1/2} S \Sigma^{-1/2}$ fall in a relatively narrow range, we can still use the Random Scaling Lemma to get a constant fraction of improvement at each iteration.

To compute these approximate diagonal values quickly, we use the random projection technique of Johnson and Lindenstrauss [JL84]. In Appendix A, we prove the following variant of their result, that deals with random projections into a space of constant dimension:

Theorem 4.3. *For all constants $\alpha, \beta, \gamma, p \in (0, 1)$, there is a positive integer $k = k_{JL}(\alpha, \beta, \gamma, p)$ such that the following holds:*

For any vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^m$ let R be a $k \times m$ matrix with entries chosen independently at random from the standard normal distribution, and let $\mathbf{w}_i = \sqrt{\frac{1}{k}} R \mathbf{v}_i$.

With probability at least p both of the following hold:

1. $\sum_{i=1}^n \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{w}_i\|^2} \leq (1 + \gamma)n$
2. $\left| \left\{ i : \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{w}_i\|^2} < 1 - \alpha \right\} \right| \leq \beta n$

Let us note that

$$S = A_2(I - A_1^T M_{11}^{-1} A_1) A_2^T = A_2(I - A_1^T M_{11}^{-1} A_1)^2 A_2^T$$

because $(I - A_1^T M_{11}^{-1} A_1)$ is a projection matrix. So if we let \mathbf{a}_i denote the i th row of A_2 , we can write the i th diagonal of S as $s_{ii} = \|(I - A_1^T M_{11}^{-1} A_1) \mathbf{a}_i^T\|^2$. Then if we use Theorem 4.3 to create a random projection matrix R , $\|R(I - A_1^T M_{11}^{-1} A_1) \mathbf{a}_i^T\|^2$ gives a good approximation to s_{ii} . Moreover, we can use one call to **ExactSolve** to compute each of the constant number of rows of the matrix $P = R(I - A_1^T M_{11}^{-1} A_1)$. Since A_2 has $\mathcal{O}(m)$ entries, we can compute PA_2^T in $\mathcal{O}(m)$ time, and obtain all the approximations $\|P \mathbf{a}_i^T\|^2$ in $\mathcal{O}(m)$ time, yielding the desired approximations of all s_{ii} values.

Our suggested algorithm, still using an exact solver, is given in Figure 1. To make this algorithm fast, we replace the calls to the exact solver with calls to the approximate solver **STSolve** of Spielman and Teng:

Theorem 4.4 (Spielman-Teng [ST04, ST06]). *The algorithm $\mathbf{x} = \text{STSolve}(M, \mathbf{b}, \epsilon)$ takes as input a symmetric diagonally-dominant $n \times n$ matrix M with m non-zeros, a column vector \mathbf{b} , and an error parameter $\epsilon > 0$, and returns in expected time $\tilde{\mathcal{O}}(m \log(1/\epsilon))$ a column vector \mathbf{x} satisfying*

$$\|\mathbf{x} - M^{-1} \mathbf{b}\|_M \leq \epsilon \|M^{-1} \mathbf{b}\|_M$$

We define the algorithm **MMatrixSolve**($A, \mathbf{b}, \epsilon, \lambda_{\min}, \lambda_{\max}$) as a modification of the algorithm **ExactMatrixSolve** in Figure 1. For this algorithm we need to provide upper and lower bounds $\lambda_{\max}, \lambda_{\min}$ on the eigenvalues of the matrix A , and the running time will depend on $\kappa = \lambda_{\max}/\lambda_{\min}$.

The modifications are that we need to set parameters:

$$\delta = (1/24) \lambda_{\min}^{1/2} \kappa^{-1/2} n^{-1} \quad \epsilon_1 = .005 (1.01 \kappa m n)^{-1/2} \quad \epsilon_2 = (1/72) \kappa^{-5/2} n^{-2}$$

and substitute the calls to **ExactSolve** in lines 2c, 2h and 3 respectively with

- **STSolve**($D_1 M_{11} D_1, D_1 A_1 \mathbf{r}_i^T, \epsilon_1$)
- **STSolve**($D_1 M_{11} D_1, D_1 (-M_{12} D_2' + \delta I) \mathbf{1}, \epsilon_2$)
- **STSolve**($D M D, D \mathbf{b}, \epsilon$).

We may note that the final call to **STSolve** guarantees that

$$\|D^{-1} \mathbf{x} - D^{-1} M^{-1} \mathbf{b}\|_{D M D} \leq \epsilon \|D^{-1} M^{-1} \mathbf{b}\|_{D M D}$$

$\mathbf{x} = \text{ExactMMatrixSolve}(A, \mathbf{b})$

Given: $n \times m$ matrix A , where $M = AA^T$ is an M-matrix and A has at most 2 non-zeros per column.

Returns: \mathbf{x} satisfying $M\mathbf{x} = \mathbf{b}$

1. Set $D := I$.
2. Until DMD is diagonally dominant do:
 - a. Permute so that $DMD = \begin{bmatrix} D_1 M_{11} D_1 & D_1 M_{12} D_2 \\ D_2 M_{12}^T D_1 & D_2 M_{22} D_2 \end{bmatrix} = \begin{bmatrix} D_1 A_1 A_1^T D_1 & D_1 A_1 A_2^T D_2 \\ D_2 A_2 A_1^T D_1 & D_2 A_2 A_2^T D_2 \end{bmatrix}$ has the diagonally dominant rows in the top section. Let $\mathbf{a}_1, \dots, \mathbf{a}_\nu$ be the rows of A_2 .
 - b. Set $k = k_{JL}(\frac{1}{100}, \frac{1}{5}, \frac{1}{100}, \frac{1}{3})$, and let R be a random $k \times m$ matrix with independent standard normal entries. Let \mathbf{r}_i be the i th row of R .
 - c. For $i = 1, \dots, k$, compute $\mathbf{q}_i^T = \text{ExactSolve}(D_1 M_{11} D_1, D_1 A_1 \mathbf{r}_i^T)$.
 - d. Set $Q = [\mathbf{q}_1^T \ \dots \ \mathbf{q}_k^T]^T$.
 - e. Let Σ be the $\nu \times \nu$ diagonal matrix with entries $\sigma_i = \|(R - Q D_1 A_1) \mathbf{a}_i^T\|^2$.
 - f. Let D_R be a uniform random $\nu \times \nu$ diagonal matrix with diagonal entries in $(0, 1)$.
 - g. Set $D'_2 = \Sigma^{-1/2} D_R$
 - h. Set D'_1 to be the matrix with diagonal $D_1 \cdot \text{ExactSolve}(D_1 M_{11} D_1, -D_1 M_{12} D'_2 \mathbf{1})$
 - i. Set $D := \begin{bmatrix} D'_1 & \\ & D'_2 \end{bmatrix}$
3. Return $\mathbf{x} = D \cdot \text{ExactSolve}(DMD, D\mathbf{b})$

Figure 1: Algorithm for solving a linear system in a symmetric M-matrix. To speed up the algorithm we will replace the exact solver with the Spielman Teng approximate solver.

or equivalently

$$\|\mathbf{x} - M^{-1}\mathbf{b}\|_M \leq \epsilon \|M^{-1}\mathbf{b}\|_M$$

so the output fulfills the specification of an approximate solver, provided that the algorithm terminates.

We can in fact bound the running time of this algorithm as follows:

Theorem 4.5. *The expected running time of the algorithm `MMatrixSolve` is $\tilde{\mathcal{O}}(m \log \frac{\kappa}{\epsilon})$.*

Proof. The running time is dominated by the calls to the Spielman-Teng solver. There are $\mathcal{O}(1)$ such solves per iterations, each of which take time $\tilde{\mathcal{O}}(m \log \kappa)$, and at the conclusion of the algorithm, there is one final call of time $\tilde{\mathcal{O}}(m \log \epsilon^{-1})$.

So, to prove the running time, it suffices for us to give a $\mathcal{O}(\log m)$ bound on the expected number of iterations. In particular, it suffices to show that in each iteration, the number of non-diagonally-dominant rows in DMD decreases by a constant fraction with constant probability.

In analyzing a single iteration, we let $D = \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix}$ denote the diagonal scaling at the start of the iteration, and we let $D' = \begin{bmatrix} D'_1 & \\ & D'_2 \end{bmatrix}$ denote the new diagonal scaling. In Appendix A, we prove:

Lemma 4.6. *D' is a positive diagonal matrix.*

This implies that $D'MD'$ has no positive off-diagonals, thereby enabling us to check which rows of $D'MD'$ are diagonally-dominant by looking for rows with nonnegative row sums.

We again let $S = M_{22} - M_{12}^T M_{11}^{-1} M_{12}$ denote the Schur complement, and let S_D denote the matrix containing the diagonal entries of S . Let us also define $\tilde{S} = \Sigma^{-1/2} S \Sigma^{-1/2}$. We know from Facts 4.1.4 and 4.1.6 that \tilde{S} is an M -matrix.

Let \tilde{S}_D be the matrix containing the diagonal entries of \tilde{S} . In Appendix A, we show that the row sums of MD' are related to \tilde{S} as follows:

Lemma 4.7.

$$MD'\mathbf{1} \geq \left[\begin{array}{c} 0 \\ \Sigma^{1/2}(\tilde{S}_D - \frac{1}{6}\tilde{S})\mathbf{1} \end{array} \right]$$

The upper part of the above inequality tells us that all the row sums that were nonnegative in DMD remain nonnegative in $D'MD'$. From the lower part of the inequality and by invoking the Random Scaling Lemma on the matrix \tilde{S} with $r = \frac{1}{6}$, we find that with probability at least $\frac{1}{23}$, the fraction of remaining rows of $D'MD'$ that now have positive row sums is at least $\frac{1}{24} \left(1 - \beta - \frac{2}{3\zeta}\right)$, where for some $\zeta < 1$, β is the fraction of the diagonal entries of \tilde{S} that are less than ζ times the average diagonal entry. Indeed we prove in Appendix A:

Lemma 4.8. *With probability at least $\frac{1}{9}$, at most $\frac{1}{5}$ of the diagonal entries of \tilde{S} are smaller than $\left(\frac{99}{101}\right)^3$ times the average diagonal entry.*

So with probability at least $\frac{1}{9} \cdot \frac{1}{23}$, the fraction of rows with negative row sums in DMD that now have positive row sums in $D'MD'$ is at least $\frac{1}{24} \left(1 - \frac{1}{5} - \frac{2}{3} \left(\frac{101}{99}\right)^3\right) > 0$.

Thus, we may conclude that `MMatrixSolve` is expected to terminate after $\mathcal{O}(\log n)$ iterations, as claimed. \square

5 Final Remarks

The reason that our interior-point algorithm currently cannot produce an exact solution to generalized flow problems is the dependence of our M -matrix solver on the condition number of the matrix, even when approximating in the matrix norm. It would be of interest to eliminate this dependence.

It would also be nice to extend the result to networks with gains. The main obstacle is that the resulting linear programs may be ill-conditioned.

References

- [AGOT92] R.K. Ahuja, A.V. Goldberg, J.B. Orlin, and R.E. Tarjan. Finding minimum-cost flows by double scaling. *Mathematical Programming*, 53:243–266, 1992.
- [AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1993.
- [Axe96] Owe Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1996.
- [BCPT05] Erik G. Boman, Doron Chen, Ojas Parekh, and Sivan Toledo. On factor width and symmetric h-matrices. *Linear Algebra and its Applications*, 405:239–248, August 2005.

- [BP94] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1994.
- [FW02] L.K. Fleischer and K.D. Wayne. Fast and simple approximation schemes for generalized flow. *Mathematical Programming*, 91(2):215–238, 2002.
- [GFNR98] T.L. Griffith, M.A. Fuller, G.B. Northcraft, and T. Radzik. Faster algorithms for the generalized network flow problem. *Mathematics of Operations Research*, 23(1):69–199, 1998.
- [GJO97] D. Goldfarb, Z. Jin, and J. Orlin. Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem. *Mathematics of Operations Research*, 22:793–802, 1997.
- [GR98] A.V. Goldberg and S. Rao. Beyond the flow decomposition barrier. *Journal of the ACM*, 1998.
- [GT87] A.V. Goldberg and R.E. Tarjan. Solving minimum cost flow problem by successive approximation. In *Proceedings of the 19th ACM Symposium on the Theory of Computing*, pages 7–18, 1987.
- [HJ91] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [JL84] W.B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26:189–206, 1984.
- [KS01] Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33th ACM Symposium on the Theory of Computing*, pages 216–223, 2001.
- [Li02] Lei Li. On the iterative criterion for generalized diagonally dominant matrices. *SIAM Journal on Matrix Analysis*, 24(1):17–24, 2002.
- [LLH⁺98] Bishan Li, Lei Li, Masunori Harada, Hiroshi Niki, and Michael J. Tsatsomeros. An iterative criterion for h-matrices. *Linear Algebra and its Applications*, 271:179–190, 1998.
- [MVV87] K. Mulmuley, U.V. Vazirani, and V.V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.
- [Orl88] J.B. Orlin. A faster strongly polynomial minimum cost flow algorithm. In *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pages 377–387, 1988.
- [Ren88] J. Renegar. A polynomial-time algorithm based on Newton’s method for linear programming. *Mathematical Programming*, 40:59–93, 1988.
- [Ren96] J. Renegar. Condition numbers, the barrier method, and the conjugate-gradient method. *SIAM Journal on Optimization*, 6:879–912, 1996.
- [Sch03] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Number 24 in Algorithms and Combinatorics. Springer-Verlag, Berlin, 2003.

- [ST04] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing (STOC-04)*, pages 81–90, New York, June 13–15 2004. ACM Press. Full version available at <http://arxiv.org/abs/cs.DS/0310051>.
- [ST06] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. Available at <http://www.arxiv.org/abs/cs.NA/0607105>, 2006.
- [TW98] E. Tardos and K.D. Wayne. Simple generalized maximum flow algorithms. In *6th International Integer Programming and Combinatorial Optimization Conference*, pages 310–324, 1998.
- [Vai89] P.M. Vaidya. Speeding up linear programming using fast matrix multiplication. In *30th Annual IEEE Symposium on Foundations of Computer Science*, pages 332–337, 1989.
- [Ye97] Y. Ye. *Interior Point Algorithms: Theory and Analysis*. John Wiley, New York, 1997.

A Proofs for Section 4

Lemma 4.3. *Given vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^m$ and constants $\alpha, \beta, \gamma, p \in (0, 1)$, for positive constant integer $k = k_{JL}(\alpha, \beta, \gamma, p)$, let R be a $k \times m$ matrix with entries chosen independently at random from the standard normal distribution, and let $\mathbf{w}_i = \sqrt{\frac{1}{k}} R \mathbf{v}_i$.*

With probability at least p both of the following hold:

- (i) $\sum_{i=1}^n \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{w}_i\|^2} \leq (1 + \gamma)n$
- (ii) $\left| \left\{ i : \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{w}_i\|^2} < 1 - \alpha \right\} \right| \leq \beta n$

Proof. Let $Z_i = \frac{\|\mathbf{v}_i\|^2}{\|\mathbf{w}_i\|^2}$ and $Z = \sum_{i=1}^n Z_i$.

Let $\mathbf{r}_1, \dots, \mathbf{r}_k$ be the rows of R , and let $w_{ij} = k^{-1/2} \langle \mathbf{r}_j, \mathbf{v}_i \rangle$ be the j th entry of \mathbf{w}_i .

Without loss of generality, we assume that all the \mathbf{v}_i are unit vectors. Thus for any given i , the expressions $k^{1/2}w_{i1}, \dots, k^{1/2}w_{ik}$ are independent standard normal random variables. So the expression

$$\frac{Z_i}{k} = \frac{1}{k \|\mathbf{w}_i\|^2} = \frac{1}{\sum_{j=1}^k (\sqrt{k}w_{ij})^2}$$

has inverse-chi-square distribution, with mean $\frac{1}{k-2}$ and variance $\frac{2}{(k-2)^2(k-4)}$. Therefore, Z has mean $\frac{kn}{k-2}$ and variance at most $\frac{2k^2n^2}{(k-2)^2(k-4)}$, because

$$\mathbf{Var}[Z] = \mathbf{Var}\left[\sum_{i=1}^n Z_i\right] = \sum_{i,j=1}^n \mathbf{Cov}[Z_i Z_j] \leq \sum_{i,j=1}^n \sqrt{\mathbf{Var}[Z_i] \mathbf{Var}[Z_j]} = n^2 \mathbf{Var}[Z_i] = k^2 n^2 \mathbf{Var}\left[\frac{Z_i}{k}\right]$$

So using Cantelli's inequality, we may conclude that

$$\mathbf{Pr}[Z > (1 + \gamma)n] < \frac{\mathbf{Var}[Z]}{\mathbf{Var}[Z] + (1 + \gamma - \frac{k}{k-2})^2 n^2} \leq \frac{2}{2 + (k-4)(1 - \frac{2}{k})^2 (\gamma - \frac{2}{k-2})^2} \quad (1)$$

By the same reasoning, $\frac{k}{Z_i}$ has chi-square distribution, with mean k and variance $2k$. So using Cantelli's inequality, we find that

$$\mathbf{Pr}[Z_i < 1 - \alpha] = \mathbf{Pr}\left[\frac{k}{Z_i} > \frac{k}{1 - \alpha}\right] < \frac{\mathbf{Var}[k/Z_i]}{\mathbf{Var}[k/Z_i] + (\frac{k}{1-\alpha} - k)^2} = \frac{2}{2 + (\frac{\alpha}{1-\alpha})^2 k}$$

Thus, the set $\{i : Z_i < 1 - \alpha\}$ has expected cardinality less than $\frac{2}{2 + (\frac{\alpha}{1-\alpha})^2 k} n$. So using Markov's inequality, we conclude that

$$\mathbf{Pr}[|\{i : Z_i < 1 - \alpha\}| > \beta n] < \frac{2}{\beta(2 + (\frac{\alpha}{1-\alpha})^2 k)} \quad (2)$$

Combining inequalities 1 and 2 via the union bound, we find the probability that (i) and (ii) both occur is at least

$$1 - \frac{2}{2 + (k-4)(1 - \frac{2}{k})^2 (\gamma - \frac{2}{k-2})^2} - \frac{2}{\beta(2 + (\frac{\alpha}{1-\alpha})^2 k)}$$

which is greater than p for sufficiently large k . \square

Lemma 4.2 (Random Scaling Lemma). *Given an $n \times n$ M -matrix M , and positive real values $\zeta \leq 1$ and $r \leq \frac{1}{4}$, let D be a random diagonal $n \times n$ matrix where each diagonal entry d_i is chosen independently and uniformly from the interval $(0, 1)$.*

Let $T \subset [n]$ be the set of rows of MD with sums at least r times the pre-scaled diagonal, i.e.

$$T = \{i \in [n] : (MD\mathbf{1})_i \geq r m_{ii}\}$$

With probability at least $\frac{1-4r}{4r+7}$, we have

$$|T| \geq \left(\frac{1}{8} - \frac{r}{2}\right) \left(1 - \beta - \frac{2}{3\zeta}\right) n$$

where β is the fraction of the diagonal entries of M that are less than ζ times the average diagonal entry.

Proof. Let M_O denote the matrix containing only the off-diagonal elements of M . Thus, M_O has no positive entries.

Let B be the set of rows of M in which the diagonal entry is less than ζ times the average diagonal entry. Thus $|B| = \beta n$.

We define a subset J of rows of M whose sums are not too far from being positive. In particular, we let J be the set of rows in which the sum of the off-diagonal entries is no less than $-\frac{3}{2}$ times the diagonal entry:

$$J = \left\{i \in [n] : (M_O\mathbf{1}_n)_i \geq -\frac{3}{2} m_{ii}\right\}$$

Let us prove that J cannot be too small. Let S be the sum of the diagonal entries of M . We have:

$$\begin{aligned} S &= \mathbf{1}_n^T M \mathbf{1}_n - \sum_{i \in [n]} (M_O\mathbf{1}_n)_i \\ &\geq - \sum_{i \in [n]} (M_O\mathbf{1}_n)_i \quad (\text{because } M \text{ is positive definite}) \\ &\geq - \sum_{i \in [n] - (J \cup B)} (M_O\mathbf{1}_n)_i \quad (\text{because } M_O \text{ is non-positive}) \\ &\geq \frac{3}{2} \sum_{i \in [n] - (J \cup B)} m_{ii} \quad (\text{by definition of } J) \\ &\geq \frac{3\zeta S}{2n} |[n] - (J \cup B)| \quad (\text{by definition of } B) \\ &\geq \frac{3\zeta S}{2n} (n - |J| - \beta n) \end{aligned}$$

So we see that $|J| \geq (1 - \beta - \frac{2}{3\zeta})n$

Next, let us show that the rows in J have a high probability of being in T . Consider the i th row sum of M_OD :

$$(M_OD\mathbf{1})_i = \sum_{j \neq i} d_j m_{ij} = \frac{1}{2} \sum_{j \neq i} m_{ij} + \sum_{j \neq i} (d_j - \frac{1}{2}) m_{ij} = \frac{1}{2} (M_O\mathbf{1})_i + \sum_{j \neq i} (d_j - \frac{1}{2}) m_{ij}$$

Since each $(d_j - \frac{1}{2})$ is symmetrically distributed around zero, we may conclude that $\frac{1}{2}(M_O\mathbf{1})_i$ is the median value of $(M_OD\mathbf{1})_i$. We may also note that $(MD\mathbf{1})_i = (M_OD\mathbf{1})_i + d_i m_{ii}$, and that the values of $(M_OD\mathbf{1})_i$ and $d_i m_{ii}$ are independent.

We thus have, for $i \in J$:

$$\begin{aligned}
\Pr[(MD\mathbf{1})_i \geq rm_{ii}] &\geq \Pr\left[(M_O D\mathbf{1})_i \geq \frac{1}{2}(M_O \mathbf{1})_i\right] \cdot \Pr\left[d_i m_{ii} \geq rm_{ii} - \frac{1}{2}(M_O \mathbf{1})_i\right] \\
&= \frac{1}{2} \cdot \Pr\left[d_i m_{ii} \geq rm_{ii} - \frac{1}{2}(M_O \mathbf{1})_i\right] \\
&\geq \frac{1}{2} \cdot \Pr\left[d_i m_{ii} \geq rm_{ii} + \frac{1}{2} \cdot \frac{3}{2} m_{ii}\right] \quad (\text{by definition of } J) \\
&= \frac{1}{2} \cdot \Pr\left[d_i \geq r + \frac{3}{4}\right] \\
&= \frac{1}{4} - r
\end{aligned}$$

Thus the expected size of $J - T$ is at most $(r + \frac{3}{4})|J|$. So we find

$$\begin{aligned}
\Pr\left[|T| > \left(\frac{1}{8} - \frac{r}{2}\right)|J|\right] &\geq \Pr\left[|J \cap T| > \left(\frac{1}{8} - \frac{r}{2}\right)|J|\right] \\
&= \Pr\left[|J - T| < \left(\frac{r}{2} + \frac{7}{8}\right)|J|\right] \\
&\geq 1 - \frac{r + \frac{3}{4}}{\frac{r}{2} + \frac{7}{8}} \quad (\text{by Markov's inequality}) \\
&= \frac{1 - 4r}{4r + 7}
\end{aligned}$$

The lemma then follows from the lower bound on $|J|$ proven above. \square

Lemma 4.6. D' is a positive diagonal matrix.

Proof. $D'_2 = \Sigma^{-1/2} D_R$ is trivially positive diagonal by construction.

To check that D'_1 is positive, we use Lemma A.1, which implies that

$$D'_1 \mathbf{1} > -M_{11}^{-1} M_{12} D'_2 \mathbf{1}_{n-\nu} + \delta \left(M_{11}^{-1} \mathbf{1}_{n-\nu} - \frac{3}{4} \lambda_{max}^{-1} \mathbf{1}_{n-\nu} \right)$$

To see why the above expression is positive, recall from Fact 4.1 that M_{11}^{-1} and $-M_{12}$ are positive matrices. Furthermore, note that the diagonals of M_{11}^{-1} are at least λ_{max}^{-1} . \square

Lemma A.1.

$$\|D'_1 \mathbf{1} - M_{11}^{-1}(-M_{12} D'_2 + \delta I) \mathbf{1}\| < \frac{3}{4} \delta \lambda_{max}^{-1}$$

Proof. Recall from the algorithm that

$$D_1^{-1} D'_1 \mathbf{1} = \text{STSolve}(D_1 M_{11} D_1, D_1(-M_{12} D'_2 + \delta I) \mathbf{1}, \epsilon_2)$$

Therefore, **STSolve** guarantees that

$$\|D_1^{-1} D'_1 \mathbf{1} - D_1^{-1} M_{11}^{-1}(-M_{12} D'_2 + \delta I) \mathbf{1}\|_{D_1 M_{11} D_1} \leq \epsilon_2 \|D_1^{-1} M_{11}^{-1}(-M_{12} D'_2 + \delta I) \mathbf{1}\|_{D_1 M_{11} D_1}$$

or equivalently

$$\|D'_1 \mathbf{1} - M_{11}^{-1}(-M_{12} D'_2 + \delta I) \mathbf{1}\|_{M_{11}} \leq \epsilon_2 \|M_{11}^{-1}(-M_{12} D'_2 \mathbf{1} + \delta I) \mathbf{1}\|_{M_{11}}$$

which in turn implies that

$$\|D'_1 \mathbf{1} - M_{11}^{-1}(-M_{12}D'_2 + \delta I)\mathbf{1}\| \leq \epsilon_2 \kappa^{1/2} \|M_{11}^{-1}(-M_{12}D'_2 + \delta I)\mathbf{1}\|$$

We can then see

$$\begin{aligned} \|D'_1 \mathbf{1} - M_{11}^{-1}(-M_{12}D'_2 + \delta I)\mathbf{1}\| &\leq \epsilon_2 \kappa^{1/2} \|-M_{11}^{-1}M_{12}D'_2 \mathbf{1} + \delta M_{11}^{-1} \mathbf{1}\| \\ &\leq \epsilon_2 \kappa^{1/2} \|M_{11}^{-1}M_{12}D'_2 \mathbf{1}\| + \delta \epsilon_2 \kappa^{1/2} \|M_{11}^{-1} \mathbf{1}\| \\ &\leq \epsilon_2 \kappa^{1/2} \|M_{11}^{-1}M_{12}D'_2 \mathbf{1}\| + \delta \epsilon_2 \kappa^{1/2} \lambda_{\min}^{-1} n^{1/2} \\ &\leq \epsilon_2 \kappa n^{1/2} \|D'_2 \mathbf{1}\| + \delta \epsilon_2 \kappa^{1/2} \lambda_{\min}^{-1} n^{1/2} \quad (\text{by Lemma A.3}) \\ &\leq \epsilon_2 \kappa n^{1/2} \|\Sigma^{-1/2} \mathbf{1}\| + \delta \epsilon_2 \kappa^{1/2} \lambda_{\min}^{-1} n^{1/2} \quad (D'_2 < \Sigma^{-1/2} \text{ by construction}) \\ &\leq 2\epsilon_2 \kappa n^{1/2} \|S_D^{-1/2} \mathbf{1}\| + \delta \epsilon_2 \kappa^{1/2} \lambda_{\min}^{-1} n^{1/2} \quad (\Sigma^{-1/2} \leq 2S_D^{-1/2} \text{ by Lemma 4.8}) \\ &\leq 2\epsilon_2 \kappa \lambda_{\min}^{-1/2} n + \delta \epsilon_2 \kappa^{1/2} \lambda_{\min}^{-1} n^{1/2} \quad (S_D^{-1/2} \mathbf{1} < \lambda_{\min}^{-1/2} \mathbf{1} \text{ by Fact 4.1.5}) \\ &= \left(2\delta^{-1} \epsilon_2 \kappa^2 \lambda_{\min}^{1/2} n + \epsilon_2 \kappa^{1/2} n^{1/2}\right) \delta \lambda_{\max}^{-1} \\ &= \left(\frac{2}{3} + \frac{1}{72} \kappa^{-2} n^{-3/2}\right) \delta \lambda_{\max}^{-1} \\ &< \frac{3}{4} \delta \lambda_{\max}^{-1} \end{aligned}$$

□

Lemma 4.8. *With probability at least $\frac{1}{9}$, at most $\frac{1}{5}$ of the diagonal entries of \tilde{S} are smaller than $(\frac{.99}{1.01})^3$ times the average diagonal entry.*

Proof. Recall that the diagonal entries of \tilde{S} are $\tilde{s}_{ii} = \frac{s_{ii}}{\sigma_i}$, where $s_{ii} = \|(I - A_1^T M_{11}^{-1} A_1) \mathbf{a}_i^T\|^2$ and $\sigma_i = \|(R - QD_1 A_1) \mathbf{a}_i^T\|^2$.

Let us define $w_i = \frac{1}{k} \|R(I - A_1^T M_{11}^{-1} A_1) \mathbf{a}_i^T\|^2$, where $k = k_{JL}(\frac{1}{100}, \frac{1}{5}, \frac{1}{100}, \frac{1}{3})$. By Lemma 4.3, there is at least $\frac{1}{3}$ probability that

$$\frac{1}{\nu} \sum_{i=1}^{\nu} \frac{s_{ii}}{w_i} \leq 1.01 \quad \text{and} \quad \left| \left\{ i : \frac{s_{ii}}{w_i} \leq .99 \right\} \right| \leq \frac{1}{5} \nu$$

So, by Lemma A.2 below, there is at least a $\frac{1}{3} - \frac{2}{9km} > \frac{1}{9}$ probability that the average diagonal entry of \tilde{S} is at most

$$\frac{1}{\nu} \sum_{i=1}^{\nu} \frac{s_{ii}}{\sigma_i} = \frac{1}{\nu} \sum_{i=1}^{\nu} \frac{s_{ii}}{w_i} \cdot \frac{w_i}{\sigma_i} \leq \frac{1.01}{k(.99)^2}$$

and similarly we have the following bound on the number of small diagonal entries:

$$\left| \left\{ i : \frac{s_{ii}}{\sigma_i} \leq \frac{.99}{k(1.01)^2} \right\} \right| \leq \frac{1}{5} \nu$$

□

Lemma A.2. *With probability at least $1 - \frac{2}{9km}$ it holds for all i that*

$$\frac{1}{k(1.01)^2} \leq \frac{w_i}{\sigma_i} \leq \frac{1}{k(.99)^2}$$

Proof. We have:

$$\begin{aligned}
\left| \sigma_i^{1/2} - k^{1/2} w_i^{1/2} \right| &= \left| \|(R - QD_1 A_1) \mathbf{a}_i^T\| - \|(R - RA_1^T M_{11}^{-1} A_1) \mathbf{a}_i^T\| \right| \\
&\leq \|((R - QD_1 A_1) - (R - RA_1^T M_{11}^{-1} A_1)) \mathbf{a}_i^T\| \\
&= \|(RA_1^T M_{11}^{-1} A_1 - QD_1 A_1) \mathbf{a}_i^T\| \\
&\leq \|\mathbf{a}_i\| \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} A_1 - \mathbf{q}_j D_1 A_1\|^2} \\
&= \|\mathbf{a}_i\| \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} D_1^{-1} - \mathbf{q}_j\|_{D_1 M_{11} D_1}^2} \\
&\leq \lambda_{\max}^{1/2} \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} D_1^{-1} - \mathbf{q}_j\|_{D_1 M_{11} D_1}^2} \\
&\quad (\|\mathbf{a}_i\|^2 \text{ is } i\text{th diagonal of } M_{22}, \text{ so cannot exceed } M_{22}\text{'s largest eigenvalue}) \\
&\leq \lambda_{\max}^{1/2} \left(\frac{s_{ii}}{\lambda_{\min}} \right)^{1/2} \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} D_1^{-1} - \mathbf{q}_j\|_{D_1 M_{11} D_1}^2} \quad (\text{using Fact 4.1.5}) \\
&= (\kappa s_{ii})^{1/2} \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} D_1^{-1} - \mathbf{q}_j\|_{D_1 M_{11} D_1}^2} \\
&\leq (\kappa s_{ii})^{1/2} \epsilon_1 \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} D_1^{-1}\|_{D_1 M_{11} D_1}^2} \quad (\text{by guarantee of STSolve}) \\
&= .005 \cdot s_{ii}^{1/2} (1.01mn)^{-1/2} \sqrt{\sum_{j=1}^k \|\mathbf{r}_j A_1^T M_{11}^{-1} A_1\|^2} \\
&= .005 \cdot s_{ii}^{1/2} (1.01mn)^{-1/2} \sqrt{\sum_{j=1}^k \|\mathbf{r}_j\|^2} \quad (\text{because } A_1^T M_{11} A_1 \text{ is a projection matrix}) \\
&\leq .01 \cdot s_{ii}^{1/2} k^{1/2} (1.01n)^{-1/2}
\end{aligned}$$

The above inequality does not hold with probability at most $\frac{2}{9km}$, based on the fact that expression $\sum_{j=1}^k \|\mathbf{r}_j\|^2$ has chi-square distribution with mk degrees of freedom.

$$\leq .01 \cdot k^{1/2} w_i^{1/2} \quad (\text{Lemma 4.3 implies that } s_{ii} \leq 1.01 \cdot n w_i)$$

So we conclude that

$$\left| \sqrt{\frac{\sigma}{w_i}} - k^{1/2} \right| \leq .01 \cdot k^{1/2}$$

□

Lemma 4.7.

$$MD' \mathbf{1} \geq \begin{bmatrix} 0 \\ \Sigma^{1/2} (\tilde{S} D_R - \frac{1}{6} \tilde{S} D) \mathbf{1} \end{bmatrix}$$

Proof.

$$\begin{aligned}
MD'\mathbf{1} &= \begin{bmatrix} M_{11}D'_1\mathbf{1} + M_{12}D'_2\mathbf{1} \\ M_{12}^TD'_1\mathbf{1} + M_{22}D'_2\mathbf{1} \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ SD'_2\mathbf{1} \end{bmatrix} + M \begin{bmatrix} D'_1\mathbf{1} + M_{11}^{-1}M_{12}D'_2\mathbf{1} - \delta M_{11}^{-1}\mathbf{1} \\ 0 \end{bmatrix} + \begin{bmatrix} \delta\mathbf{1} \\ \delta M_{12}^TM_{11}^{-1}\mathbf{1} \end{bmatrix} \\
&\geq \begin{bmatrix} 0 \\ SD'_2\mathbf{1} \end{bmatrix} - \lambda_{\max} \|D'_1\mathbf{1} + M_{11}^{-1}M_{12}D'_2\mathbf{1} - \delta M_{11}^{-1}\mathbf{1}\| \mathbf{1} + \begin{bmatrix} \delta\mathbf{1} \\ -\delta\|M_{12}^TM_{11}^{-1}\mathbf{1}\|\mathbf{1} \end{bmatrix} \\
&\geq \begin{bmatrix} 0 \\ SD'_2\mathbf{1} \end{bmatrix} - \frac{3}{4}\delta\mathbf{1} + \begin{bmatrix} \delta\mathbf{1} \\ -\delta\kappa^{1/2}n\mathbf{1} \end{bmatrix} \quad (\text{using Lemmas A.1 and A.3}) \\
&\geq \begin{bmatrix} 0 \\ SD'_2\mathbf{1} - 2\delta\kappa^{1/2}n\mathbf{1} \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ S\Sigma^{-1/2}D_R\mathbf{1} - \frac{1}{12}\lambda_{\min}^{1/2}\mathbf{1} \end{bmatrix} \\
&\geq \begin{bmatrix} 0 \\ S\Sigma^{-1/2}D_R\mathbf{1} - \frac{1}{12}S_D^{1/2}\mathbf{1} \end{bmatrix} \quad (\text{using Fact 4.1.5}) \\
&\geq \begin{bmatrix} 0 \\ S\Sigma^{-1/2}D_R\mathbf{1} - \frac{1}{6}S_D^{1/2}\tilde{S}_D^{1/2}\mathbf{1} \end{bmatrix} \quad (\text{using Lemma 4.8})
\end{aligned}$$

□

Lemma A.3. *For all positive vectors \mathbf{v} ,*

$$\|M_{12}^TM_{11}^{-1}\mathbf{v}\| \leq \kappa^{1/2}n^{1/2}\|\mathbf{v}\|$$

Proof. Define $c = \lambda_{\min}^{-1}\kappa^{-1/2}n^{-1/2}\|\mathbf{v}\| = \lambda_{\max}^{-1}\kappa^{1/2}n^{-1/2}\|\mathbf{v}\|$.

$$\begin{aligned}
\|M_{12}^TM_{11}^{-1}\mathbf{v}\| &\leq \|M_{12}^TM_{11}^{-1}\mathbf{v}\|_1 = -\mathbf{1}^TM_{12}^TM_{11}^{-1}\mathbf{v} \quad (\text{by Fact 4.1, } M_{11}^{-1} \text{ and } -M_{12} \text{ are nonnegative}) \\
&= \frac{1}{2c} \left(\mathbf{v}^TM_{11}^{-1}\mathbf{v} + c^2\mathbf{1}^TM_{22}\mathbf{1} - [\mathbf{v}^TM_{11}^{-1} \quad c\mathbf{1}^T] M \begin{bmatrix} M_{11}^{-1}\mathbf{v} \\ c\mathbf{1} \end{bmatrix} \right) \\
&\leq \frac{1}{2c} (\mathbf{v}^TM_{11}^{-1}\mathbf{v} + c^2\mathbf{1}^TM_{22}\mathbf{1}) \\
&\leq \frac{1}{2} \left(\frac{\|\mathbf{v}\|^2}{c\lambda_{\min}} + c\lambda_{\max}n \right) = \kappa^{1/2}n^{1/2}\|\mathbf{v}\|
\end{aligned}$$

□

B Solving Matrices from the Interior-Point Method

In the interior-point algorithm, we need to solve matrices of the form

$$M + \mathbf{v}\mathbf{v}^T = \begin{bmatrix} AD_1^2 A^T + D_2^2 & AD_1^2 \\ D_1^2 A^T & D_1^2 + D_3^2 \end{bmatrix} + \mathbf{v}\mathbf{v}^T$$

where A is an $n \times m$ matrix with entries bounded by U in absolute value, AA^T is an M-matrix, and D_1, D_2, D_3 are positive diagonal matrices. We show how to do this using our `MMatrixSolve` algorithm.

Consider the Schur complement of M :

$$M_S = (AD_1^2 A^T + D_2^2) - AD_1^2 (D_1^2 + D_3^2)^{-1} D_1^2 A^T = AD_1^2 D_3^2 (D_1^2 + D_3^2)^{-1} A^T + D_2^2 = A_S A_S^T$$

where $A_S = [AD_1 D_3 (D_1^2 + D_3^2)^{-1/2} \quad D_2]$. Note that M_S is also an M-matrix, and that the eigenvalues of M_S fall in the range $[d_{min}^2, d_{max}^2 (U\sqrt{nm} + 1)]$ where d_{min} and d_{max} are respectively the smallest and largest diagonal entry in D_1, D_2, D_3 .

We can build an solver for systems in M from a solver for systems in M_S , by using the following easily verifiable property of the Schur complement:

Lemma B.1. For $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^T & M_{22} \end{bmatrix}$ and Schur complement $M_S = M_{11} - M_{12} M_{22}^{-1} M_{12}^T$, we have

$$\left\| \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} - M^{-1} \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \right\|_M = \left\| \mathbf{x}_1 - M_S^{-1} (\mathbf{b}_1 - M_{12} M_{22}^{-1} \mathbf{b}_2) \right\|_{M_S} + \left\| \mathbf{x}_2 - M_{22}^{-1} (\mathbf{b}_2 - M_{12}^T \mathbf{x}_1) \right\|_{M_{22}}$$

Then, to solve systems in $M + \mathbf{v}\mathbf{v}^T$, we can use the Sherman-Morrison formula:

$$(M + \mathbf{v}\mathbf{v}^T)^{-1} = M^{-1} - \frac{M^{-1} \mathbf{v}\mathbf{v}^T M^{-1}}{1 + \mathbf{v}^T M^{-1} \mathbf{v}}$$

In particular, we give the following algorithm, which runs in time $\tilde{O}\left(m \log \frac{\kappa \|\mathbf{v}\|}{\epsilon}\right)$:

$\mathbf{x} = \text{Solve}(M + \mathbf{v}\mathbf{v}^T, \mathbf{b}, \epsilon)$ where $M = \begin{bmatrix} AD_1^2 A^T + D_2^2 & AD_1^2 \\ D_1^2 A^T & D_1^2 + D_3^2 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$ and $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$

- Define $\epsilon_1 = \frac{\epsilon}{2}(1 + \mathbf{v}^T M^{-1} \mathbf{v})^{-1}$ and $\epsilon_2 = \min\left\{\frac{1}{2}, \frac{\epsilon}{14}(1 + \mathbf{v}^T M^{-1} \mathbf{v})^{-1}\right\}$
- $\mathbf{y}' = \text{MMatrixSolve}(A_S, \mathbf{b}_1 - AD_1^2 (D_1^2 + D_3^2)^{-1} \mathbf{b}_2, \epsilon_1, d_{min}^2, d_{max}^2 (U\sqrt{nm} + 1))$
- $\mathbf{y} = \begin{bmatrix} \mathbf{y}' \\ (D_1^2 + D_3^2)^{-1} (\mathbf{b}_2 - D_1^2 A^T \mathbf{y}') \end{bmatrix}$
- $\mathbf{z}' = \text{MMatrixSolve}(A_S, \mathbf{v}_1 - AD_1^2 (D_1^2 + D_3^2)^{-1} \mathbf{v}_2, \epsilon_2, d_{min}^2, d_{max}^2 (U\sqrt{nm} + 1))$
- $\mathbf{z} = \begin{bmatrix} \mathbf{z}' \\ (D_1^2 + D_3^2)^{-1} (\mathbf{v}_2 - D_1^2 A^T \mathbf{z}') \end{bmatrix}$
- Return $\mathbf{x} = \mathbf{y} - \frac{\mathbf{z}\mathbf{z}^T \mathbf{b}}{1 + \mathbf{v}^T \mathbf{z}}$

Lemma B.2. $\mathbf{x} = \text{Solve}(M + \mathbf{v}\mathbf{v}^T, \mathbf{b}, \epsilon)$ satisfies

$$\|\mathbf{x} - (M + \mathbf{v}\mathbf{v}^T)^{-1}\mathbf{b}\|_{M+\mathbf{v}\mathbf{v}^T} < \epsilon \|(M + \mathbf{v}\mathbf{v}^T)^{-1}\mathbf{b}\|_{M+\mathbf{v}\mathbf{v}^T}$$

Proof. We first show that $\|\mathbf{y} - M^{-1}\mathbf{b}\|_M \leq \epsilon_1 \|M^{-1}\mathbf{b}\|_M$:

$$\begin{aligned} \|\mathbf{y} - M^{-1}\mathbf{b}\|_M &= \|\mathbf{y}' - M_S^{-1}(\mathbf{b}_1 - AD_1^2(D_1^2 + D_3^2)^{-1}\mathbf{b}_2)\|_{M_S} \quad (\text{by Lemma B.1}) \\ &\leq \epsilon_1 \|M_S^{-1}(\mathbf{b}_1 - AD_1^2(D_1^2 + D_3^2)^{-1}\mathbf{b}_2)\|_{M_S} \quad (\text{guaranteed by MMatrixSolve}) \\ &= \epsilon_1 \left(\|M^{-1}\mathbf{b}\|_M - \|M_{22}^{-1}\mathbf{b}_2\|_{M_{22}} \right) \quad (\text{by Lemma B.1}) \\ &\leq \epsilon_1 \|M^{-1}\mathbf{b}\|_M \end{aligned}$$

By the same reasoning, $\|\mathbf{z} - M^{-1}\mathbf{v}\|_M \leq \epsilon_2 \|M^{-1}\mathbf{v}\|_M$.

Next, let us define the inner product $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_M = \mathbf{v}_1^T M \mathbf{v}_2$. We will use repeatedly the inequality $|\langle \mathbf{v}_1, \mathbf{v}_2 \rangle_M| \leq \|\mathbf{v}_1\|_M \|\mathbf{v}_2\|_M$

Recall that we return the value $\mathbf{x} = \mathbf{y} - \frac{\mathbf{z}\mathbf{z}^T\mathbf{b}}{1+\mathbf{v}^T\mathbf{z}}$. So we begin by analyzing the expressions $\mathbf{z}\mathbf{z}^T\mathbf{b}$ and $\mathbf{v}^T\mathbf{z}$:

$$\begin{aligned} \|\mathbf{z}\mathbf{z}^T\mathbf{b} - M^{-1}\mathbf{v}\mathbf{v}^T M^{-1}\mathbf{b}\|_M &\leq \|\mathbf{z}\mathbf{z}^T\mathbf{b} - \mathbf{z}\mathbf{v}^T M^{-1}\mathbf{b}\|_M + \|\mathbf{z}\mathbf{v}^T M^{-1}\mathbf{b} - M^{-1}\mathbf{v}\mathbf{v}^T M^{-1}\mathbf{b}\|_M \\ &= |\langle \mathbf{z} - M^{-1}\mathbf{v}, M^{-1}\mathbf{b} \rangle_M| \|\mathbf{z}\|_M + |\langle M^{-1}\mathbf{v}, M^{-1}\mathbf{b} \rangle_M| \|\mathbf{z} - M^{-1}\mathbf{v}\|_M \\ &\leq \|\mathbf{z} - M^{-1}\mathbf{v}\|_M \|M^{-1}\mathbf{b}\|_M \|\mathbf{z}\|_M + \|M^{-1}\mathbf{v}\|_M \|M^{-1}\mathbf{b}\|_M \|\mathbf{z} - M^{-1}\mathbf{v}\|_M \\ &= \|\mathbf{z} - M^{-1}\mathbf{v}\|_M \|M^{-1}\mathbf{b}\|_M (\|\mathbf{z}\|_M + \|M^{-1}\mathbf{v}\|_M) \\ &\leq \|\mathbf{z} - M^{-1}\mathbf{v}\|_M \|M^{-1}\mathbf{b}\|_M (\|\mathbf{z} - M^{-1}\mathbf{v}\|_M + 2\|M^{-1}\mathbf{v}\|_M) \\ &\leq \epsilon_2(\epsilon_2 + 2) \|M^{-1}\mathbf{v}\|_M^2 \|M^{-1}\mathbf{b}\|_M \end{aligned} \tag{3}$$

$$\begin{aligned} |\mathbf{v}^T\mathbf{z} - \mathbf{v}^T M^{-1}\mathbf{v}| &= |\langle M^{-1}\mathbf{v}, \mathbf{z} - M^{-1}\mathbf{v} \rangle_M| \\ &\leq \|M^{-1}\mathbf{v}\|_M \|\mathbf{z} - M^{-1}\mathbf{v}\|_M \\ &\leq \epsilon_2 \|M^{-1}\mathbf{v}\|_M^2 \\ &= \epsilon_2(\mathbf{v}^T M^{-1}\mathbf{v}) \end{aligned} \tag{4}$$

We thus have:

$$\begin{aligned}
& \| \mathbf{x} - (M + \mathbf{v}\mathbf{v}^T)^{-1} \mathbf{b} \|_M \tag{5} \\
&= \left\| \left(\mathbf{y} - \frac{\mathbf{z}\mathbf{z}^T \mathbf{b}}{1 + \mathbf{v}^T \mathbf{z}} \right) - \left(M^{-1} \mathbf{b} - \frac{M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b}}{1 + \mathbf{v}^T M^{-1} \mathbf{v}} \right) \right\|_M \\
&\leq \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \left\| \frac{\mathbf{z}\mathbf{z}^T \mathbf{b}}{1 + \mathbf{v}^T \mathbf{z}} - \frac{M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b}}{1 + \mathbf{v}^T \mathbf{z}} \right\|_M + \left\| \frac{M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b}}{1 + \mathbf{v}^T \mathbf{z}} - \frac{M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b}}{1 + \mathbf{v}^T M^{-1} \mathbf{v}} \right\|_M \\
&= \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \frac{1}{1 + \mathbf{v}^T \mathbf{z}} \left(\| \mathbf{z}\mathbf{z}^T \mathbf{b} - M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b} \|_M + \frac{|\mathbf{v}^T \mathbf{z} - \mathbf{v}^T M^{-1} \mathbf{v}|}{1 + \mathbf{v}^T M^{-1} \mathbf{v}} \| M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b} \|_M \right) \\
&\leq \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \frac{1}{\mathbf{v}^T \mathbf{z}} \left(\| \mathbf{z}\mathbf{z}^T \mathbf{b} - M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b} \|_M + \frac{|\mathbf{v}^T \mathbf{z} - \mathbf{v}^T M^{-1} \mathbf{v}|}{\mathbf{v}^T M^{-1} \mathbf{v}} \| M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b} \|_M \right) \\
&\leq \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \frac{1}{(1 - \epsilon_2) \mathbf{v}^T M^{-1} \mathbf{v}} \left(\epsilon_2 (\epsilon_2 + 2) \| M^{-1} \mathbf{v} \|_M^2 \| M^{-1} \mathbf{b} \|_M + \epsilon_2 \| M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{b} \|_M \right) \\
&\quad (\text{by equations 3 and 4}) \\
&= \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \frac{\epsilon_2 (\epsilon_2 + 2) \| M^{-1} \mathbf{v} \|_M^2 \| M^{-1} \mathbf{b} \|_M + \epsilon_2 |\langle M^{-1} \mathbf{v}, M^{-1} \mathbf{b} \rangle_M| \| M^{-1} \mathbf{v} \|_M}{(1 - \epsilon_2) \| M^{-1} \mathbf{v} \|_M^2} \\
&\leq \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \frac{\epsilon_2 (\epsilon_2 + 2) \| M^{-1} \mathbf{v} \|_M^2 \| M^{-1} \mathbf{b} \|_M + \epsilon_2 \| M^{-1} \mathbf{v} \|_M^2 \| M^{-1} \mathbf{b} \|_M}{(1 - \epsilon_2) \| M^{-1} \mathbf{v} \|_M^2} \\
&= \| \mathbf{y} - M^{-1} \mathbf{b} \|_M + \frac{\epsilon_2 (\epsilon_2 + 3)}{1 - \epsilon_2} \| M^{-1} \mathbf{b} \|_M \\
&\leq \left(\epsilon_1 + \frac{\epsilon_2 (\epsilon_2 + 3)}{1 - \epsilon_2} \right) \| M^{-1} \mathbf{b} \|_M \\
&\leq \epsilon (1 + \mathbf{v}^T M^{-1} \mathbf{v})^{-1} \| M^{-1} \mathbf{b} \|_M \tag{6}
\end{aligned}$$

So we conclude

$$\begin{aligned}
\| \mathbf{x} - (M + \mathbf{v}\mathbf{v}^T)^{-1} \mathbf{b} \|_{M+\mathbf{v}\mathbf{v}^T} &\leq (1 + \mathbf{v}^T M^{-1} \mathbf{v})^{1/2} \| \mathbf{x} - (M + \mathbf{v}\mathbf{v}^T)^{-1} \mathbf{b} \|_M \quad \text{by Lemma B.3(i)} \\
&\leq \epsilon (1 + \mathbf{v}^T M^{-1} \mathbf{v})^{-1/2} \| M^{-1} \mathbf{b} \|_M \quad \text{by equation (6)} \\
&= \epsilon (1 + \mathbf{v}^T M^{-1} \mathbf{v})^{-1/2} \| \mathbf{b} \|_{M^{-1}} \\
&\leq \epsilon \| \mathbf{b} \|_{(M+\mathbf{v}\mathbf{v}^T)^{-1}} \quad \text{by Lemma B.3(ii)} \\
&= \epsilon \| (M + \mathbf{v}\mathbf{v}^T)^{-1} \mathbf{b} \|_{M+\mathbf{v}\mathbf{v}^T}
\end{aligned}$$

□

Lemma B.3. For all vectors \mathbf{v} , \mathbf{w} , and symmetric positive definite M :

$$\begin{aligned}
(i) \quad & \| \mathbf{w} \|_{M+\mathbf{v}\mathbf{v}^T} \leq \| \mathbf{w} \|_M (1 + \mathbf{v}^T M^{-1} \mathbf{v})^{1/2} \\
(ii) \quad & \| \mathbf{w} \|_{(M+\mathbf{v}\mathbf{v}^T)^{-1}} \geq \| \mathbf{w} \|_{M^{-1}} (1 + \mathbf{v}^T M^{-1} \mathbf{v})^{-1/2}
\end{aligned}$$

Proof of (i).

$$\begin{aligned}
\|\mathbf{w}\|_{M+\mathbf{v}\mathbf{v}^T} &= (\mathbf{w}^T(M+\mathbf{v}\mathbf{v}^T)\mathbf{w})^{1/2} \\
&= (\mathbf{w}^T M \mathbf{w} + (\mathbf{w}^T \mathbf{v})^2)^{1/2} \\
&= (\|\mathbf{w}\|_M^2 + \langle \mathbf{w}, M^{-1} \mathbf{v} \rangle_M^2)^{1/2} \\
&\leq (\|\mathbf{w}\|_M^2 + \|\mathbf{w}\|_M^2 \|M^{-1} \mathbf{v}\|_M^2)^{1/2} \\
&= \|\mathbf{w}\|_M (1 + \|M^{-1} \mathbf{v}\|_M^2)^{1/2}
\end{aligned}$$

□

Proof of (ii).

$$\begin{aligned}
\|\mathbf{w}\|_{(M+\mathbf{v}\mathbf{v}^T)^{-1}} &= (\mathbf{w}^T(M+\mathbf{v}\mathbf{v}^T)^{-1}\mathbf{w})^{1/2} \\
&= \left(\mathbf{w}^T \left(M^{-1} - \frac{M^{-1} \mathbf{v} \mathbf{v}^T M^{-1}}{1 + \mathbf{v}^T M^{-1} \mathbf{v}} \right) \mathbf{w} \right)^{1/2} \\
&= \left(\mathbf{w}^T M^{-1} \mathbf{w} - \frac{\mathbf{w}^T M^{-1} \mathbf{v} \mathbf{v}^T M^{-1} \mathbf{w}}{1 + \mathbf{v}^T M^{-1} \mathbf{v}} \right)^{1/2} \\
&= \left(\|\mathbf{w}\|_{M^{-1}}^2 - \frac{\langle \mathbf{w}, \mathbf{v} \rangle_{M^{-1}}^2}{1 + \|\mathbf{v}\|_{M^{-1}}^2} \right)^{1/2} \\
&\geq \left(\|\mathbf{w}\|_{M^{-1}}^2 - \frac{\|\mathbf{w}\|_{M^{-1}}^2 \|\mathbf{v}\|_{M^{-1}}^2}{1 + \|\mathbf{v}\|_{M^{-1}}^2} \right)^{1/2} \\
&= \|\mathbf{w}\|_{M^{-1}} \left(1 - \frac{\|\mathbf{v}\|_{M^{-1}}^2}{1 + \|\mathbf{v}\|_{M^{-1}}^2} \right)^{1/2} \\
&= \|\mathbf{w}\|_{M^{-1}} \left(1 + \|\mathbf{v}\|_{M^{-1}}^2 \right)^{-1/2}
\end{aligned}$$

□

C Interior-Point Method using an Approximate Solver

Throughout this section, we take `Solve` to be an algorithm such that $\mathbf{x} = \text{Solve}(M, \mathbf{b}, \epsilon)$ satisfies

$$\|\mathbf{x} - M^{-1}\mathbf{b}\|_M \leq \epsilon \|M^{-1}\mathbf{b}\|$$

We use the notational convention that S denotes the diagonal matrix whose diagonal is \mathbf{s} . The same applies for X and \mathbf{x} , etc.

$\mathbf{1}_k$ denotes the all-ones vector of length k .

$\overset{\circ}{\Omega}$ denotes the interior of polytope Ω .

We are given a canonical primal linear program

$$\mathbf{z}^* = \min_{\mathbf{x}} \{ \mathbf{c}^T \mathbf{x} : A\mathbf{x} = \mathbf{b}; \mathbf{x} \geq 0 \}$$

which has the same solution as the dual linear program

$$\mathbf{z}^* = \max_{(\mathbf{y}, \mathbf{s})} \{ \mathbf{b}^T \mathbf{y} : A^T \mathbf{y} + \mathbf{s} = \mathbf{c}; \mathbf{s} \geq 0 \}$$

where A is an $n \times m$ matrix, $\mathbf{x}, \mathbf{s}, \mathbf{c}$ are length m , and \mathbf{y}, \mathbf{b} are length n , and $m \geq n$. (Unfortunately, this use of n and m is reversed from the standard linear programming convention. We do this to be consistent with the standard graph-theory convention that we use throughout the paper.)

We let Ω^D denote the dual polytope

$$\Omega^D = \{ (\mathbf{y}, \mathbf{s}) : A^T \mathbf{y} + \mathbf{s} = \mathbf{c}; \mathbf{s} \geq 0 \}$$

so we can write the solution to the linear program as $\mathbf{z}^* = \max_{(\mathbf{y}, \mathbf{s}) \in \Omega^D} \mathbf{b}^T \mathbf{y}$.

In this appendix, we present an `InteriorPoint` algorithm based on that of Renegar [Ren88], modified to use an approximate solver. Our analysis follows that found in [Ye97].

Theorem 2.1. $\mathbf{x} = \text{InteriorPoint}(A, \mathbf{b}, \mathbf{c}, \lambda_{\min}, T, \mathbf{y}^0, \epsilon)$ takes input that satisfy

- A is an $n \times m$ matrix; \mathbf{b} is a length n vector; \mathbf{c} is a length m vector
- AA^T is positive definite, and $\lambda_{\min} > 0$ is a lower bound on the eigenvalues of AA^T
- $T > 0$ is an upper bound on the absolute values of the dual coordinates, i.e.
 $\|\mathbf{y}\|_{\infty} < T$ and $\|\mathbf{s}\|_{\infty} < T$ for all (\mathbf{y}, \mathbf{s}) that satisfy $\mathbf{s} = \mathbf{c} - A^T \mathbf{y} \geq 0$
- initial point \mathbf{y}^0 is a length n vector where $A^T \mathbf{y}^0 < \mathbf{c}$
- error parameter ϵ satisfies $0 < \epsilon < 1$

and returns $\mathbf{x} > 0$ satisfying $\|A\mathbf{x} - \mathbf{b}\| \leq \epsilon$ and $z^* < \mathbf{c}^T \mathbf{x} < z^* + \epsilon$.

Let us define

- U is the largest absolute value of any entry in $A, \mathbf{b}, \mathbf{c}$
- s_{\min}^0 is the smallest entry of $\mathbf{s}^0 = \mathbf{c} - A^T \mathbf{y}^0$

Then the algorithm makes $\mathcal{O}\left(\sqrt{m} \log \frac{TUm}{\lambda_{\min} s_{\min}^0 \epsilon}\right)$ calls to the approximate solver, of the form

$$\text{Solve}(AS^{-2}A^T + \mathbf{v}\mathbf{v}^T, \cdot, \epsilon')$$

where S is a positive diagonal matrix with condition number $\mathcal{O}\left(\frac{T^2 U m^2}{\epsilon}\right)$, and \mathbf{v}, ϵ' satisfy

$$\log \frac{\|\mathbf{v}\|}{\epsilon'} = \mathcal{O}\left(\log \frac{TUm}{s_{\min}^0 \epsilon}\right)$$

C.1 The Analytic Center

Standard interior-point methods focus on a particular point in the interior of the dual polytope. This point, called the *analytic center*, is the point that maximizes the product of the slacks, i.e. the product of the elements of \mathbf{s} . For the purpose of our analysis, we use the following equivalent definition of the analytic center:

Fact C.1 (see [Ye97, §3.1]). *The analytic center of $\Omega^D = \{(\mathbf{y}, \mathbf{s}) : A^T \mathbf{y} + \mathbf{s} = \mathbf{c}; \mathbf{s} \geq 0\}$ is the unique point $(\mathbf{y}^*, \mathbf{s}^*) \in \mathring{\Omega}^D$ that satisfies $\eta_A(\mathbf{s}^*) = 0$, where we define*

$$\begin{aligned}\mathbf{x}_A(\mathbf{s}) &= S^{-1}(I - S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1})\mathbf{1}_m \\ \eta_A(\mathbf{s}) &= \|S\mathbf{x}_A(\mathbf{s}) - \mathbf{1}_m\| = \|S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}_m\| = \|AS^{-1}\mathbf{1}_m\|_{(AS^{-2}A^T)^{-1}}\end{aligned}$$

These definitions of \mathbf{x}_A and η_A satisfying the following properties:

Lemma C.2. *Let $(\mathbf{y}^*, \mathbf{s}^*)$ be the analytic center of Ω^D . For any point $(\mathbf{y}, \mathbf{s}) \in \mathring{\Omega}^D$ we have*

- (i) $A\mathbf{x}_A(\mathbf{s}) = 0$
- (ii) $\eta_A(\mathbf{s}) < 1$ implies $\mathbf{x}_A(\mathbf{s}) > 0$
- (iii) $\mathbf{x}_A(\mathbf{s}^*) = S^{-1}\mathbf{1}_m$
- (iv) For all \mathbf{x} satisfying $A\mathbf{x} = 0$, it holds that $\|S\mathbf{x} - \mathbf{1}_m\| \geq \eta_A(\mathbf{s})$

The first three properties are straightforward from the definition. We present a proof of the last:

Proof of C.2(iv). Note that $S\mathbf{x}_A(\mathbf{s}) - \mathbf{1}_m$ is orthogonal to $S(\mathbf{x} - \mathbf{x}_A(\mathbf{s}))$, because

$$\begin{aligned}\langle S\mathbf{x}_A(\mathbf{s}) - \mathbf{1}_m, S(\mathbf{x} - \mathbf{x}_A(\mathbf{s})) \rangle &= \langle S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}_m, S(\mathbf{x} - \mathbf{x}_A(\mathbf{s})) \rangle \\ &= \langle (AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}_m, A(\mathbf{x} - \mathbf{x}_A(\mathbf{s})) \rangle \\ &= \langle (AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}_m, 0 \rangle \\ &= 0\end{aligned}$$

We thus have

$$\|S\mathbf{x} - \mathbf{1}_m\| = \|S\mathbf{x}_A(\mathbf{s}) - \mathbf{1}_m + S(\mathbf{x} - \mathbf{x}_A(\mathbf{s}))\| \geq \|S\mathbf{x}_A(\mathbf{s}) - \mathbf{1}_m\| = \eta_A(\mathbf{s})$$

□

It will be useful to note that the slacks of the analytic center cannot be too small. We can bound the slacks of the analytic center away from zero as follows:

Lemma C.3 (compare [Ye97, Thm 2.6]). *Let $(\mathbf{y}^*, \mathbf{s}^*)$ be the analytic center of Ω^D . For every $(\mathbf{y}, \mathbf{s}) \in \Omega^D$, we have $\mathbf{s}^* > \frac{1}{m}\mathbf{s}$*

Proof.

$$\begin{aligned}\|S^{-1}\mathbf{s}\|_\infty &\leq \mathbf{1}_m^T S^{-1}\mathbf{s} = \mathbf{1}_m^T S^{-1}\mathbf{s}^* + \mathbf{1}_m^T S^{-1}(\mathbf{s} - \mathbf{s}^*) \\ &= m + \mathbf{1}_m^T S^{-1}(\mathbf{s} - \mathbf{s}^*) \\ &= m + \mathbf{1}_m^T S^{-1}((\mathbf{c} - A^T \mathbf{y}) - (\mathbf{c} - A^T \mathbf{y}^*)) \\ &= m + \mathbf{1}_m^T S^{-1}A^T(\mathbf{y}^* - \mathbf{y}) \\ &= m\end{aligned}$$

where we know from Lemmas C.2(i) and C.2(iii) that $AS^{-1}\mathbf{1}_m = 0$

□

$\mathbf{y}^+ = \text{NewtonStep}(A, \mathbf{c}, \mathbf{y})$

- Let $\mathbf{s} = \mathbf{c} - A^T \mathbf{y}$
- Let $\mathbf{d}_y = \text{Solve}(AS^{-2}A^T, -AS^{-1}\mathbf{1}_m, \epsilon_3)$ where $\epsilon_3 = \frac{1}{20(\sqrt{m+1})}$
- Return $\mathbf{y}^+ = \mathbf{y} + (1 - \epsilon_3)\mathbf{d}_y$

Figure 2: Procedure for stepping closer to the analytic center

Let us note that a point $(\mathbf{y}, \mathbf{s}) \in \mathring{\Omega}^D$ that satisfies $\eta_A(\mathbf{s}) < 1$ is close to the analytic center, in the sense that the slacks \mathbf{s} are bounded by a constant ratio from the slacks of the analytic center:

Lemma C.4 ([Ye97, Thm 3.2(iv)]). *Suppose $(\mathbf{y}, \mathbf{s}) \in \mathring{\Omega}^D$ satisfies $\eta_A(\mathbf{s}) = \eta < 1$ and let $(\mathbf{y}^*, \mathbf{s}^*)$ be the analytic center of Ω^D . Then $\|S^{-1}\mathbf{s}^* - \mathbf{1}_m\| \leq \frac{\eta}{1-\eta}$.*

If $(\mathbf{y}, \mathbf{s}) \in \mathring{\Omega}^D$ is sufficiently close to the analytic center (as measured by η_A), then with a single call to the approximate solver, we can take a Newton-type step to find a point even closer to the analytic center. This **NewtonStep** procedure is presented in Figure 2.

In the first part of the following lemma, we prove that the point returned by **NewtonStep** is indeed still inside the dual polytope. In the second part, we show how close the new point is to the analytic center:

Lemma C.5 (compare [Ye97, Thm 3.3]). *Suppose $(\mathbf{y}, \mathbf{s}) \in \mathring{\Omega}^D$ satisfies $\eta_A(\mathbf{s}) = \eta < 1$.*

Let $\mathbf{y}^+ = \text{NewtonStep}(A, \mathbf{c}, \mathbf{y})$ and $\mathbf{s}^+ = \mathbf{c} - A^T \mathbf{y}^+$

Then (i) $\mathbf{s}^+ > 0$ and (ii) $\eta_A(\mathbf{s}^+) \leq \eta^2 + \frac{1}{20}\eta$

Proof. (i) The solver guarantees that

$$\|\mathbf{d}_y + (AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}\|_{AS^{-2}A^T} \leq \epsilon_3 \|(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}\|_{AS^{-2}A^T} = \epsilon_3 \cdot \eta$$

or equivalently

$$\|S^{-1}A^T\mathbf{d}_y + S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}\| \leq \epsilon_3 \|S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}\| = \epsilon_3 \cdot \eta \quad (7)$$

and so

$$\|S^{-1}A^T\mathbf{d}_y\| \leq (1 + \epsilon_3) \|S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}\| = (1 + \epsilon_3)\eta < 1 + \epsilon_3$$

We thus have

$$\begin{aligned} \|S^{-1}\mathbf{s}^+ - \mathbf{1}\| &= \|S^{-1}(\mathbf{s} - (1 - \epsilon_3)A^T\mathbf{d}_y) - \mathbf{1}\| \\ &= (1 - \epsilon_3) \|S^{-1}A^T\mathbf{d}_y\| \\ &\leq (1 - \epsilon_3)(1 + \epsilon_3) < 1 \end{aligned}$$

Thus $S^{-1}\mathbf{s}^+$ is positive and so is \mathbf{s}^+ .

(ii) Let $\mathbf{x} = \mathbf{x}_A(\mathbf{s})$ and $\mathbf{x}^+ = \mathbf{x}_A(\mathbf{s}^+)$. We have

$$\begin{aligned}
\eta_A(\mathbf{s}^+) &\leq \|X\mathbf{s}^+ - \mathbf{1}_m\| \quad (\text{by Lemma C.2(iv)}) \\
&= \|X(\mathbf{s} - (1 - \epsilon_3)A^T\mathbf{d}_y) - \mathbf{1}_m\| \\
&= \|(1 - \epsilon_3)XS(X\mathbf{s} - \mathbf{1}_m - S^{-1}A^T\mathbf{d}_y) - (1 - \epsilon_3)(XS - I)(X\mathbf{s} - \mathbf{1}_m) + \epsilon_3(X\mathbf{s} - \mathbf{1}_m)\| \\
&\leq (1 - \epsilon_3)\|XS(S^{-1}A^T\mathbf{d}_y - S\mathbf{x} + \mathbf{1}_m)\| + (1 - \epsilon_3)\|(XS - I)(X\mathbf{s} - \mathbf{1}_m)\| + \epsilon_3\|(X\mathbf{s} - \mathbf{1}_m)\| \\
&\leq (1 - \epsilon_3)\|S\mathbf{x}\| \|S^{-1}A^T\mathbf{d}_y - S\mathbf{x} + \mathbf{1}_m\| + (1 - \epsilon_3)\|S\mathbf{x} - \mathbf{1}_m\|^2 + \epsilon_3\|S\mathbf{x} - \mathbf{1}_m\| \\
&\quad (\text{using the relation } \|V\mathbf{w}\| \leq \|\mathbf{v}\|_\infty \|\mathbf{w}\| \leq \|\mathbf{v}\| \|\mathbf{w}\|) \\
&\leq (1 - \epsilon_3)(\|S\mathbf{x} - \mathbf{1}_m\| + \|\mathbf{1}_m\|) \|S^{-1}A^T\mathbf{d}_y - S\mathbf{x} + \mathbf{1}_m\| + (1 - \epsilon_3)\|S\mathbf{x} - \mathbf{1}_m\|^2 + \epsilon_3\|S\mathbf{x} - \mathbf{1}_m\| \\
&= (1 - \epsilon_3)(\eta + \sqrt{m}) \|S^{-1}A^T\mathbf{d}_y - S\mathbf{x} + \mathbf{1}_m\| + (1 - \epsilon_3)\eta^2 + \epsilon_3\eta \\
&= (1 - \epsilon_3)(\eta + \sqrt{m}) \|S^{-1}A^T\mathbf{d}_y + S^{-1}A^T(AS^{-2}A^T)^{-1}AS^{-1}\mathbf{1}_m\| + (1 - \epsilon_3)\eta^2 + \epsilon_3\eta \\
&\leq (1 - \epsilon_3)(\eta + \sqrt{m})\epsilon_3\eta + (1 - \epsilon_3)\eta^2 + \epsilon_3\eta \quad (\text{by equation 7}) \\
&\leq \epsilon_3(\eta + \sqrt{m})\eta + (1 - \epsilon_3)\eta^2 + \epsilon_3\eta \\
&= \eta^2 + \epsilon_3(\sqrt{m} + 1)\eta \\
&= \eta^2 + \frac{1}{20}\eta
\end{aligned}$$

□

C.2 The Path-Following Algorithm

In a path-following algorithm, we modify the dual polytope $\Omega^D = \{(\mathbf{y}, \mathbf{s}) : A^T\mathbf{y} + \mathbf{s} = \mathbf{c}; \mathbf{s} \geq 0\}$ by adding an additional constraint $\mathbf{b}^T\mathbf{y} \geq z$, where $z \leq z^*$. As we let z approach z^* , the center of the polytope approaches the solution to the dual linear program.

Letting $s_{\text{gap}} = \mathbf{b}^T\mathbf{y} - z$ denote the new slack variable, we define the modified polytope:

$$\Omega_{\mathbf{b},z}^D = \left\{ (\mathbf{y}, \mathbf{s}, s_{\text{gap}}) : \begin{bmatrix} A^T\mathbf{y} + \mathbf{s} \\ -\mathbf{b}^T\mathbf{y} + s_{\text{gap}} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ -z \end{bmatrix}; \mathbf{s}, s_{\text{gap}} \geq 0 \right\}$$

Using a trick of Renegar, when we define the analytic center of $\Omega_{\mathbf{b},z}^D$, we consider there to be m copies of the slack s_{gap} , as follows:

Definition C.6. The **analytic center** of $\Omega_{\mathbf{b},z}^D$ is the point $(\mathbf{y}^*, \mathbf{s}^*, s_{\text{gap}}^*) \in \mathring{\Omega}_{\mathbf{b},z}^D$, that satisfies $\tilde{\eta}(\mathbf{s}^*, s_{\text{gap}}^*) = 0$, where we define

$$\tilde{\eta}(\mathbf{s}, s_{\text{gap}}) = \eta_{\tilde{A}}(\tilde{\mathbf{s}}) \quad \text{where } \tilde{A} = [A \quad -\mathbf{b}\mathbf{1}_m^T] \text{ and } \tilde{\mathbf{s}} = \begin{bmatrix} \mathbf{s} \\ s_{\text{gap}}\mathbf{1}_m \end{bmatrix}$$

The **central path** is the set of analytic centers of the polytopes $\left\{ \Omega_{\mathbf{b},z}^D \right\}_{z \leq z^*}$

A path-following algorithm steps through a sequence of points near the central path, as z increases towards z^* . It is useful to note that given any point on the central path, we may easily construct a feasible primal solution \mathbf{x} , as follows:

Lemma C.7. Let $(\mathbf{y}^*, \mathbf{s}^*, s_{\text{gap}}^*)$ be the analytic center of $\Omega_{\mathbf{b},z}^D$. Then the vector $\mathbf{x} = \frac{s_{\text{gap}}^*}{m} \mathring{S}^{-1}\mathbf{1}_m$ satisfies $A\mathbf{x} = \mathbf{b}$. More generally, for any $(\mathbf{y}, \mathbf{s}, s_{\text{gap}}) \in \mathring{\Omega}_{\mathbf{b},z}^D$, the vector $\mathbf{x} = \frac{s_{\text{gap}}}{m} S^{-1}\mathbf{1}_m$ satisfies

$$\|A\mathbf{x} - \mathbf{b}\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} = \frac{s_{\text{gap}}}{m} \cdot \tilde{\eta}(\mathbf{s}, s_{\text{gap}})$$

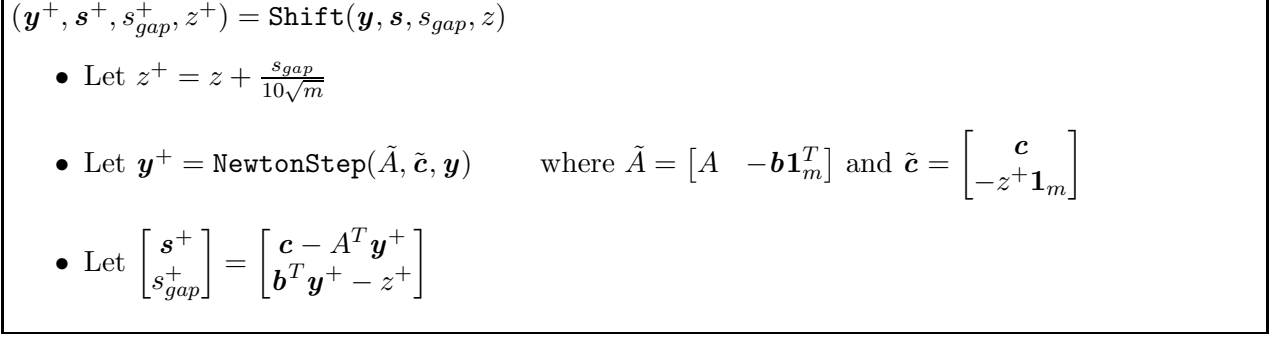


Figure 3: Procedure for taking a step along the central path

Proof. We prove the second assertion:

$$\begin{aligned}
\|A\mathbf{x} - \mathbf{b}\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} &= \frac{s_{gap}}{m} \|AS^{-1}\mathbf{1}_m - ms_{gap}^{-1}\mathbf{b}\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} \\
&= \frac{s_{gap}}{m} \|\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m}\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} \\
&= \frac{s_{gap}}{m} \cdot \tilde{\eta}(\mathbf{s}, s_{gap})
\end{aligned}$$

The first assertion now follows from the definition of analytic center. \square

Let us now describe how to take steps along the central path using our approximate solver. In Figure 3, we present the procedure **Shift**, which takes as input a value $z < z^*$ and a point $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \mathring{\Omega}_{\mathbf{b},z}^D$ satisfying $\eta(\mathbf{s}, s_{gap}) \leq \frac{1}{10}$. The output is a new value z^+ that is closer to z^* , and a new point $(\mathbf{y}^+, \mathbf{s}^+, s_{gap}^+) \in \mathring{\Omega}_{\mathbf{b},z^+}^D$ satisfying $\eta(\mathbf{s}^+, s_{gap}^+) \leq \frac{1}{10}$. The procedure requires a single call to the solver.

Let us examine this procedure more closely. After defining the incremented value z^+ , if we let $s'_{gap} = \mathbf{b}^T \mathbf{y} - z^+ = s_{gap} - (z^+ - z)$, then $(\mathbf{y}, \mathbf{s}, s'_{gap})$ is a point in the shifted polytope $\Omega_{\mathbf{b},z^+}^D$. However this point may be slightly farther away from the central path. One call to the **NewtonStep** procedure suffices to obtain a new point $(\mathbf{y}^+, \mathbf{s}^+, s_{gap}^+) \in \mathring{\Omega}_{\mathbf{b},z^+}^D$ that is sufficiently close to the central path, satisfying $\tilde{\eta}(\mathbf{s}^+, s_{gap}^+) \leq \frac{1}{10}$.

We prove this formally:

Lemma C.8 (compare [Ye97, Lem 4.5]). *Given $z < z^*$ and $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \mathring{\Omega}_{\mathbf{b},z}^D$ where $\tilde{\eta}(\mathbf{s}, s_{gap}) \leq \frac{1}{10}$, let $s'_{gap} = \mathbf{b}^T \mathbf{y} - z^+$ and $(\mathbf{y}^+, \mathbf{s}^+, s_{gap}^+, z^+) = \text{Shift}(\mathbf{y}, \mathbf{s}, s_{gap}, z)$. Then*

- (i) $z^+ < z^*$
- (ii) $s'_{gap} > 0$ and $\tilde{\eta}(\mathbf{s}, s'_{gap}) < \frac{21}{100}$
- (iii) $\mathbf{s}^+, s_{gap}^+ > 0$ and $\tilde{\eta}(\mathbf{s}^+, s_{gap}^+) < \frac{1}{10}$

Proof. (i)

$$z^+ = z + \frac{\mathbf{b}^T \mathbf{y} - z}{10\sqrt{m}} < z + (\mathbf{b}^T \mathbf{y} - z) = \mathbf{b}^T \mathbf{y} < z^*$$

(ii) We note that $s'_{gap} = s_{gap} - (z^+ - z) = \left(1 - \frac{1}{10\sqrt{m}}\right) s_{gap} > 0$.

Let us write $\tilde{\mathbf{s}} = \begin{bmatrix} \mathbf{s} \\ s_{gap} \mathbf{1}_m \end{bmatrix}$ and $\tilde{\mathbf{s}}' = \begin{bmatrix} \mathbf{s} \\ s'_{gap} \mathbf{1}_m \end{bmatrix}$ and note that

$$\tilde{\mathbf{s}} - \tilde{\mathbf{s}}' = \begin{bmatrix} 0 \\ (s_{gap} - s'_{gap}) \mathbf{1}_m \end{bmatrix} = \begin{bmatrix} 0 \\ (z^+ - z) \mathbf{1}_m \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{s_{gap}}{10\sqrt{m}} \mathbf{1}_m \end{bmatrix} \quad (8)$$

Let us define $\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{gap} \end{bmatrix} = \mathbf{x}_{\tilde{A}}(\tilde{\mathbf{s}})$. So we have

$$\begin{aligned} \tilde{\eta}(\mathbf{s}, s'_{gap}) &= \eta_{\tilde{A}}(\tilde{\mathbf{s}}') \leq \left\| \tilde{S}' \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| \quad (\text{by Lemma C.2(iv)}) \\ &\leq \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \left\| (\tilde{S}' - \tilde{S}) \tilde{\mathbf{x}} \right\| \\ &= \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \frac{1}{10\sqrt{m}} \|s_{gap} \mathbf{x}_{gap}\| \quad (\text{by Equation 8}) \\ &\leq \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \frac{1}{10\sqrt{m}} (\|s_{gap} \mathbf{x}_{gap} - \mathbf{1}_m\| + \|\mathbf{1}_m\|) \\ &= \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \frac{1}{10\sqrt{m}} \|s_{gap} \mathbf{x}_{gap} - \mathbf{1}_m\| + \frac{1}{10} \\ &\leq \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \frac{1}{10} \|s_{gap} \mathbf{x}_{gap} - \mathbf{1}_m\| + \frac{1}{10} \\ &\leq \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \frac{1}{10} \left\| \tilde{S} \tilde{\mathbf{x}} - \mathbf{1}_{2m} \right\| + \frac{1}{10} \\ &= \frac{11}{10} \tilde{\eta}(\mathbf{s}, s_{gap}) + \frac{1}{10} \\ &\leq \frac{11}{10} \cdot \frac{1}{10} + \frac{1}{10} = \frac{21}{100} \end{aligned} \quad (9)$$

(iii) By Lemma C.5, we have $\mathbf{s}^+, s_{gap}^+ > 0$ and

$$\tilde{\eta}(\mathbf{s}^+, s_{gap}^+) \leq \tilde{\eta}(\mathbf{s}, s'_{gap})^2 + \frac{1}{20} \tilde{\eta}(\mathbf{s}, s'_{gap}) \leq \left(\frac{21}{100} \right)^2 + \frac{1}{20} \cdot \frac{21}{100} < \frac{1}{10}$$

□

We now present the complete path-following **InteriorPoint** algorithm, implemented using an approximate solver, in Figure 4. For now we postpone describing the **FindCentralPath** subroutine, which gives an initial point near the central path. In particular, it produces a $z^C < z^*$ and $(\mathbf{y}^C, \mathbf{s}^C, s_{gap}^C) \in \Omega_{\mathbf{b}, z^C}^D$ satisfying $\tilde{\eta}(\mathbf{s}^C, s_{gap}^C) \leq \frac{1}{10}$. Once we have this initial central path point, Lemma C.8 tells us that after each call to **Shift** we have a new value $z < z^*$ and new central path point $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \hat{\Omega}_{\mathbf{b}, z}^D$ that satisfies $\tilde{\eta}(\mathbf{s}, s_{gap}) \leq \frac{1}{10}$.

Later we will analyze the number of calls to **Shift** before the algorithm terminates. First, let us confirm that the algorithm returns the correct output:

Lemma C.9. *The output of $\mathbf{x} = \text{InteriorPoint}(A, \mathbf{b}, \mathbf{c}, \mathbf{y}^0, \epsilon)$ satisfies*

- (i) $\mathbf{x} > 0$
- (ii) $\|A\mathbf{x} - \mathbf{b}\| \leq \frac{\epsilon}{12\sqrt{2} \cdot T n^{1/2}}$
- (iii) $\mathbf{c}^T \mathbf{x} < z^* + \epsilon$

$\mathbf{x} = \text{InteriorPoint}(A, \mathbf{b}, \mathbf{c}, \mathbf{y}^0, \epsilon)$

- Compute $(\mathbf{y}^C, z^C) = \text{FindCentralPath}(A, \mathbf{b}, \mathbf{c}, \mathbf{y}^0)$ and $\begin{bmatrix} \mathbf{s}^C \\ s_{gap}^C \end{bmatrix} = \begin{bmatrix} \mathbf{c} - A^T \mathbf{y}^C \\ \mathbf{b}^T \mathbf{y}^C - z^C \end{bmatrix}$
- Set $(\mathbf{y}, \mathbf{s}, s_{gap}, z) := (\mathbf{y}^C, \mathbf{s}^C, s_{gap}^C, z^C)$
- While $s_{gap} > \frac{\epsilon}{3}$:
 - Set $(\mathbf{y}, \mathbf{s}, s_{gap}, z) := \text{Shift}(\mathbf{y}, \mathbf{s}, s_{gap}, z)$
- Compute $\mathbf{v} = \text{Solve}(\tilde{A}\tilde{S}^{-2}\tilde{A}^T, \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m}, \epsilon_4)$

$$\text{where } \tilde{A} = \begin{bmatrix} A & -\mathbf{b}\mathbf{1}_m^T \end{bmatrix} \quad \text{and } \epsilon_4 = \min \left(1, \frac{s_{min}}{TU} \cdot \frac{m^{1/2}}{n} \right)$$

$$\text{and } \tilde{\mathbf{s}} = \begin{bmatrix} \mathbf{s} \\ s_{gap}\mathbf{1}_m \end{bmatrix} \quad \text{and } s_{min} \text{ is the smallest entry of } \tilde{\mathbf{s}}$$

- Return $\mathbf{x} = \frac{\mathbf{x}'}{mx'_{gap}}$ where $\begin{bmatrix} \mathbf{x}' \\ x'_{gap} \end{bmatrix} = \begin{bmatrix} S^{-1}\mathbf{1}_m - S^{-2}A^T\mathbf{v} \\ s_{gap}^{-1} + s_{gap}^{-2}\mathbf{b}^T\mathbf{v} \end{bmatrix}$

Figure 4: Dual path-following interior-point algorithm using an approximate solver

Proof. (i) To assist in our proof, let us define $\tilde{\mathbf{x}}' = \begin{bmatrix} \mathbf{x}' \\ x'_{gap}\mathbf{1}_m \end{bmatrix}$ and note that $\tilde{\mathbf{x}}' = \tilde{S}^{-1}\mathbf{1}_{2m} - \tilde{S}^{-2}\tilde{A}^T\mathbf{v}$. We have

$$\begin{aligned} \|\tilde{S}\tilde{\mathbf{x}}' - \mathbf{1}_{2m}\| &= \|\tilde{S}^{-1}\tilde{A}^T\mathbf{v}\| \\ &= \|\mathbf{v}\|_{\tilde{A}\tilde{S}^{-2}\tilde{A}^T} \\ &\leq \left\| (\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{\tilde{A}\tilde{S}^{-2}\tilde{A}^T} + \left\| \mathbf{v} - (\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{\tilde{A}\tilde{S}^{-2}\tilde{A}^T} \\ &\leq (1 + \epsilon_4) \left\| (\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{\tilde{A}\tilde{S}^{-2}\tilde{A}^T} \quad (\text{by guarantee of Solve}) \\ &= (1 + \epsilon_4) \cdot \tilde{\eta}(\mathbf{s}, s_{gap}) \leq 2 \cdot \tilde{\eta}(\mathbf{s}, s_{gap}) \leq 2 \cdot \frac{1}{10} \leq \frac{1}{5} \end{aligned} \tag{10}$$

Since $\tilde{\mathbf{s}}$ is positive, we conclude that $\tilde{\mathbf{x}}'$ must also be positive, and so must be \mathbf{x} .

(ii) We have

$$\begin{aligned} \|A\mathbf{x} - \mathbf{b}\| &= \frac{1}{mx'_{gap}} \|A\mathbf{x}' - mx'_{gap}\mathbf{b}\| \\ &= \frac{1}{mx'_{gap}} \|\tilde{A}\tilde{\mathbf{x}}'\| \\ &= \frac{1}{mx'_{gap}} \|\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} - \tilde{A}\tilde{S}^{-2}\tilde{A}^T\mathbf{v}\| \end{aligned}$$

Observe that the largest eigenvalue of the matrix $\tilde{A}\tilde{A}^T = AA^T + m\mathbf{b}\mathbf{b}^T$ is less than the trace, which is at most $2nmU^2$. Thus, the largest eigenvalue of $\tilde{A}\tilde{S}^{-2}\tilde{A}^T$ is at most $2nmU^2s_{\min}^{-2}$. So we proceed

$$\begin{aligned}
&\leq \frac{(2nmU^2s_{\min}^{-2})^{1/2}}{mx'_{gap}} \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} - \tilde{A}\tilde{S}^{-2}\tilde{A}^T\mathbf{v} \right\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} \\
&= \frac{(2n)^{1/2}U}{m^{1/2}x'_{gap}s_{\min}} \left\| (\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} - \mathbf{v} \right\|_{\tilde{A}\tilde{S}^{-2}\tilde{A}^T} \\
&\leq \frac{(2n)^{1/2}U}{m^{1/2}x'_{gap}s_{\min}} \cdot \epsilon_4 \left\| (\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}\tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{\tilde{A}\tilde{S}^{-2}\tilde{A}^T} \quad (\text{by guarantee of Solve}) \\
&= \frac{(2n)^{1/2}U}{m^{1/2}x'_{gap}s_{\min}} \cdot \epsilon_4 \cdot \tilde{\eta}(\mathbf{s}, s_{gap}) \\
&\leq \frac{(2n)^{1/2}U}{m^{1/2}x'_{gap}s_{\min}} \cdot \epsilon_4 \cdot \frac{1}{10} \\
&\leq \frac{(2n)^{1/2}U}{m^{1/2}x'_{gap}s_{\min}} \cdot \frac{m^{1/2}s_{\min}}{nTU} \cdot \frac{1}{10} \\
&= \frac{1}{5\sqrt{2} \cdot Tn^{1/2}} \cdot \frac{1}{x'_{gap}} \\
&\leq \frac{1}{5\sqrt{2} \cdot Tn^{1/2}} \cdot \frac{5}{4} \cdot s_{gap} \quad (\text{from equation 10, we know } s_{gap}x'_{gap} \geq \frac{4}{5}) \\
&\leq \frac{1}{5\sqrt{2} \cdot Tn^{1/2}} \cdot \frac{5}{4} \cdot \frac{\epsilon}{3} \\
&\leq \frac{\epsilon}{12\sqrt{2} \cdot Tn^{1/2}}
\end{aligned} \tag{11}$$

(iii) We have

$$\begin{aligned}
\mathbf{s}^T \mathbf{x} &= \frac{\mathbf{s}^T \mathbf{x}'}{mx'_{gap}} \\
&\geq \frac{5\mathbf{s}^T \mathbf{x}'}{6m} \cdot s_{gap} \quad (\text{from equation 10, we know } s_{gap}x'_{gap} \leq \frac{6}{5}) \\
&= \frac{5\|\mathbf{S}\mathbf{x}'\|_1}{6m} \cdot s_{gap} \\
&\geq \frac{5}{6m} (\|\mathbf{1}_m\|_1 - \|\mathbf{S}\mathbf{x}' - \mathbf{1}_m\|_1) \cdot s_{gap} \\
&= \frac{5}{6m} (m - \|\mathbf{S}\mathbf{x}' - \mathbf{1}_m\|_1) \cdot s_{gap} \\
&\geq \frac{5}{6m} (m - \sqrt{m}\|\mathbf{S}\mathbf{x}' - \mathbf{1}_m\|) \cdot s_{gap} \\
&\geq \frac{5}{6m} \left(m - \frac{1}{5}\sqrt{m} \right) \cdot s_{gap} \quad (\text{by equation 10}) \\
&\geq \frac{5}{6m} \cdot \frac{4}{5} \cdot m \cdot s_{gap} \\
&= \frac{2}{3} \cdot s_{gap}
\end{aligned} \tag{12}$$

$$\begin{aligned}
\mathbf{s}^T \mathbf{x} &= \frac{\mathbf{s}^T \mathbf{x}'}{mx'_{gap}} \\
&\leq \frac{5\mathbf{s}^T \mathbf{x}'}{4m} \cdot s_{gap} \quad (\text{from equation 10, we know } s_{gap}x'_{gap} \geq \frac{4}{5}) \\
&= \frac{5\|S\mathbf{x}'\|_1}{4m} \cdot s_{gap} \\
&\leq \frac{5}{4m} (\|S\mathbf{x}' - \mathbf{1}_m\|_1 + \|\mathbf{1}_m\|_1) \cdot s_{gap} \\
&= \frac{5}{4m} (\|S\mathbf{x}' - \mathbf{1}_m\|_1 + m) \cdot s_{gap} \\
&\leq \frac{5}{4m} (\sqrt{m}\|S\mathbf{x}' - \mathbf{1}_m\| + m) \cdot s_{gap} \\
&\leq \frac{5}{4m} \left(\frac{1}{5}\sqrt{m} + m \right) \cdot s_{gap} \quad (\text{by equation 10}) \\
&\leq \frac{5}{4m} \cdot \frac{6}{5} \cdot m \cdot s_{gap} \\
&= \frac{3}{2} \cdot s_{gap}
\end{aligned} \tag{13}$$

We then have

$$\begin{aligned}
\mathbf{c}^T \mathbf{x} - z^* &> \mathbf{c}^T \mathbf{x} - \mathbf{b}^T \mathbf{y} \\
&= (\mathbf{c}^T - \mathbf{y}^T A) \mathbf{x} + \mathbf{y}^T (A\mathbf{x} - \mathbf{b}) \\
&= \mathbf{s}^T \mathbf{x} + \mathbf{y}^T (A\mathbf{x} - \mathbf{b}) && \geq \frac{2}{3} \cdot s_{gap} + \mathbf{y}^T (A\mathbf{x} - \mathbf{b}) \quad (\text{by Equation 13}) \\
&\leq \dots \frac{2}{3} \cdot \epsilon + \mathbf{y}^T (A\mathbf{x} - \mathbf{b}) \\
&\leq \frac{2}{3} \cdot \epsilon + \|\mathbf{y}\| \|A\mathbf{x} - \mathbf{b}\| \\
&\leq \frac{2}{3} \cdot \epsilon + Tn^{1/2} \|A\mathbf{x} - \mathbf{b}\| \\
&\leq \frac{2}{3} \cdot \epsilon + \frac{1}{12\sqrt{2}} \cdot \epsilon \quad (\text{by Lemma C.9(ii)}) \\
&< \epsilon
\end{aligned}$$

$$\begin{aligned}
\mathbf{c}^T \mathbf{x} - z^* &< \mathbf{c}^T \mathbf{x} - z \\
&= (\mathbf{c}^T - \mathbf{y}^T A) \mathbf{x} + \mathbf{y}^T (A \mathbf{x} - \mathbf{b}) + \mathbf{b}^T \mathbf{y} - z \\
&= \mathbf{s}^T \mathbf{x} + \mathbf{y}^T (A \mathbf{x} - \mathbf{b}) + s_{gap} \\
&\leq \frac{5}{2} \cdot s_{gap} + \mathbf{y}^T (A \mathbf{x} - \mathbf{b}) \quad (\text{by Equation 13}) \\
&\leq \frac{5}{6} \cdot \epsilon + \mathbf{y}^T (A \mathbf{x} - \mathbf{b}) \\
&\leq \frac{5}{6} \cdot \epsilon + \|\mathbf{y}\| \|A \mathbf{x} - \mathbf{b}\| \\
&\leq \frac{5}{6} \cdot \epsilon + T n^{1/2} \|A \mathbf{x} - \mathbf{b}\| \\
&\leq \frac{5}{6} \cdot \epsilon + \frac{1}{12\sqrt{2}} \cdot \epsilon \quad (\text{by Lemma C.9(ii)}) \\
&< \epsilon
\end{aligned}$$

□

Next, we analyze the number of **Shift** iterations until the algorithm terminates. We can measure the progress of the algorithm with the potential function $B(z)$:

$$B(z) = \sum_{j=1}^m \log \hat{\mathbf{s}}_j + m \log \hat{s}_{gap} \quad \text{where } (\hat{\mathbf{y}}, \hat{\mathbf{s}}, \hat{s}_{gap}) \text{ is the analytic center of } \Omega_{\mathbf{b},z}^D$$

Soon, we will show how a decrease in $B(z)$ implies that s_{gap} is decreasing and thus the algorithm is making progress. Let us first show that the value of $B(z)$ decreases by $\Omega(\sqrt{m})$ after each iteration.

Lemma C.10 (compare [Ye97, Lem 4.6]). *Given $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \mathring{\Omega}_{\mathbf{b},z}^D$ satisfying $\tilde{\eta}(\mathbf{s}, s_{gap}) < \frac{1}{10}$, let $(\mathbf{y}^+, z^+) = \text{Shift}(\mathbf{y}, z)$. Then $B(z^+) \leq B(z) - \Theta(\sqrt{m})$.*

Proof. Let $(\hat{\mathbf{y}}, \hat{\mathbf{s}}, \hat{s}_{gap})$ and $(\hat{\mathbf{y}}^+, \hat{\mathbf{s}}^+, \hat{s}_{gap}^+)$ respectively be the analytic centers of $\Omega_{\mathbf{b},z}^D$ and $\Omega_{\mathbf{b},z^+}^D$.

Following Lemma C.7, we define $\mathbf{x} = \frac{s_{gap}^*}{m} \dot{S}^{-1} \mathbf{1}_m$ that satisfies $A\mathbf{x} = \mathbf{b}$. We have

$$\begin{aligned}
e^{\frac{B(z^+) - B(z)}{2m}} &= \sqrt{\left(\prod_{j=1}^m \frac{s_j^+}{s_j^*} \right)^{\frac{1}{m}} \cdot \frac{s_{gap}^+}{s_{gap}^*}} \\
&\leq \frac{1}{2m} \sum_{j=1}^m \frac{s_j^+}{s_j^*} + \frac{1}{2} \frac{s_{gap}^+}{s_{gap}^*} \\
&\leq 1 + \frac{1}{2m} \sum_{j=1}^m \frac{s_j^+ - s_j^*}{s_j^*} + \frac{1}{2} \frac{s_{gap}^+ - s_{gap}^*}{s_{gap}^*} \\
&= 1 + \frac{1}{2s_{gap}^*} \cdot \frac{s_{gap}^*}{m} (\mathbf{s}^+ - \mathbf{s})^T \dot{S}^{-1} \mathbf{1}_m + \frac{1}{2s_{gap}^*} (s_{gap}^+ - s_{gap}^*) \\
&= 1 + \frac{1}{2s_{gap}^*} ((\mathbf{s}^+ - \mathbf{s})^T \mathbf{x} + (s_{gap}^+ - s_{gap}^*)) \\
&= 1 + \frac{1}{2s_{gap}^*} \left(((\mathbf{c} - A^T \mathbf{y}^+) - (\mathbf{c} - A^T \mathbf{y}))^T \mathbf{x} + (s_{gap}^+ - s_{gap}^*) \right) \\
&= 1 + \frac{1}{2s_{gap}^*} ((\mathbf{y} - \mathbf{y}^+)^T A\mathbf{x} + (s_{gap}^+ - s_{gap}^*)) \\
&= 1 + \frac{1}{2s_{gap}^*} ((\mathbf{y} - \mathbf{y}^+)^T \mathbf{b} + (s_{gap}^+ - s_{gap}^*)) \quad (\text{by Lemma C.7}) \\
&= 1 + \frac{1}{2s_{gap}^*} ((\mathbf{b}^T \mathbf{y} - s_{gap}^*) - (\mathbf{b}^T \mathbf{y}^+ - s_{gap}^*)) \\
&= 1 + \frac{1}{2s_{gap}^*} (z - z^+) \\
&= 1 - \frac{s_{gap}}{20\sqrt{m} \cdot s_{gap}^*} \\
&\leq 1 - \frac{9}{200\sqrt{m}} \quad \left(\frac{s_{gap}}{s_{gap}^*} \leq \frac{10}{9} \text{ by Lemma C.4} \right) \\
&\leq e^{-\frac{9}{200\sqrt{m}}}
\end{aligned} \tag{14}$$

We conclude that $B(z^+) - B(z) \leq -\frac{9}{100}\sqrt{m}$. \square

Let us now show that a decrease in the potential function $B(z)$ implies a decrease in the value of s_{gap} :

Lemma C.11 (compare [Ye97, Prop 4.2]). *Given $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \mathring{\Omega}_{\mathbf{b}, z}^D$ and $(\mathbf{y}^+, \mathbf{s}^+, s_{gap}^+) \in \mathring{\Omega}_{\mathbf{b}, z^+}^D$ where $z^+ > z$ and $\tilde{\eta}(\mathbf{s}, s_{gap}) \leq \eta$ and $\tilde{\eta}(\mathbf{s}^+, s_{gap}^+) \leq \eta$ for $\eta < 1$. Then*

$$\frac{s_{gap}}{s_{gap}^+} \leq (1 - 2\eta) \cdot \left(e^{\frac{B(z_1) - B(z_2)}{m}} - 1 \right)$$

Proof. Let $(\mathbf{y}^*, \mathbf{s}^*, s_{gap}^*)$ and $(\mathbf{y}^+, \mathbf{s}^+, s_{gap}^+)$ respectively be the analytic centers of $\Omega_{\mathbf{b}, z}^D$ and $\Omega_{\mathbf{b}, z^+}^D$. We define $\mathbf{x} = \frac{s_{gap}^*}{m} \dot{S}^{-1} \mathbf{1}_m$ and $\mathbf{x}^+ = \frac{s_{gap}^+}{m} (\dot{S}^+)^{-1} \mathbf{1}_m$, which by Lemma C.7 satisfy $A\mathbf{x} = \mathbf{b} = A\mathbf{x}^+$.

We have

$$\begin{aligned}
e^{\frac{B(z) - B(z^+)}{m}} &= \left(\prod_{j=1}^m \frac{\mathbf{s}_j}{\mathbf{s}_j^+} \right)^{\frac{1}{m}} \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&\leq \frac{1}{m} \left(\sum_{j=1}^m \frac{\mathbf{s}_j}{\mathbf{s}_j^+} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{m} \sum_{j=1}^m \frac{\mathbf{s}_j - \mathbf{s}_j^+}{\mathbf{s}_j^+} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} (\mathbf{s} - \mathbf{s}^+)^T \mathbf{x}^+ \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} ((\mathbf{c} - A^T \mathbf{y}) - (\mathbf{c} - A^T \mathbf{y}^+))^T \mathbf{x}^+ \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} (\mathbf{y} - \mathbf{y}^+)^T A \mathbf{x}^+ \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} (\mathbf{y} - \mathbf{y}^+)^T \mathbf{b} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} (\mathbf{y} - \mathbf{y}^+)^T A \mathbf{x} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} ((\mathbf{c} - A^T \mathbf{y}) - (\mathbf{c} - A^T \mathbf{y}^+))^T \mathbf{x} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{1}{\mathbf{s}_{gap}^+} (\mathbf{s} - \mathbf{s}^+)^T \mathbf{x} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&\leq \left(1 + \frac{1}{\mathbf{s}_{gap}^+} \mathbf{s}^T \mathbf{x} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&= \left(1 + \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \right) \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \\
&\leq \left(1 + \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \right)^2
\end{aligned}$$

So

$$\frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \geq e^{\frac{B(z) - B(z^+)}{2m}} - 1$$

Using Lemma C.4, we may conclude

$$\frac{s_{gap}}{s_{gap}^+} = \frac{s_{gap}}{\mathbf{s}_{gap}} \cdot \frac{\mathbf{s}_{gap}}{\mathbf{s}_{gap}^+} \cdot \frac{\mathbf{s}_{gap}^+}{s_{gap}^+} \geq \frac{1 - \frac{\eta}{1-\eta}}{1 + \frac{\eta}{1-\eta}} \cdot \left(e^{\frac{B(z) - B(z^+)}{2m}} - 1 \right)$$

□

Corollary C.12. *The InteriorPoint algorithm makes $\mathcal{O}\left(\sqrt{m} \log \frac{s_{gap}^C}{\epsilon}\right)$ calls to Shift.*

Proof. Recall that the algorithm will terminate only when the value of s_{gap} has decreased from its initial value of s_{gap}^C to below $\frac{\epsilon}{3}$. Thus, Lemma C.11 ensures us that s_{gap} will be smaller than $\frac{\epsilon}{3}$ once $B(z)$ has decreased by $\Omega\left(m \log \frac{s_{gap}^C}{\epsilon}\right)$. According to Lemma C.10, this occurs after $\mathcal{O}\left(\sqrt{m} \log \frac{s_{gap}^C}{\epsilon}\right)$ **Shift** iterations. \square

C.3 Finding the Central Path

It remains for us to describe how to initialize the path-following algorithm by finding a point near the central path. Essentially, this is accomplished by running the path-following algorithm in reverse. Instead of stepping towards the optimum given by \mathbf{b} , we step away from the optimum given by the vector $\underline{\mathbf{b}} = A(S^0)^{-1}\mathbf{1}_m$ that depends on our initial feasible point $(\mathbf{y}^0, \mathbf{s}^0) \in \mathring{\Omega}^D$.

Our analysis parallels that in the previous section. The following function $\underline{\eta}$ measures the proximity of a point $(\mathbf{y}, \mathbf{s}, \underline{s}_{gap}) \in \mathring{\Omega}_{\underline{\mathbf{b}}, z}^D$ to the central path given by $\underline{\mathbf{b}}$:

$$\underline{\eta}(\mathbf{s}, \underline{s}_{gap}) = \eta_{\tilde{A}}(\tilde{\mathbf{s}}) \quad \text{where } \tilde{A} = \begin{bmatrix} A & -\underline{\mathbf{b}}\mathbf{1}_m^T \end{bmatrix} \text{ and } \tilde{\mathbf{s}} = \begin{bmatrix} \mathbf{s} \\ \underline{s}_{gap}\mathbf{1}_m \end{bmatrix}$$

To initialize the algorithm, we observe that $(\mathbf{y}^0, \mathbf{s}^0, m) \in \Omega_{\underline{\mathbf{b}}, z^0}^D$ is on the $\underline{\mathbf{b}}$ central path, where we define $z^0 = \underline{\mathbf{b}}^T \mathbf{y}^0 - m$:

Lemma C.13. $\underline{\eta}(\mathbf{s}^0, m) = 0$

Proof. Defining $\tilde{\mathbf{s}}^0 = \begin{bmatrix} \mathbf{s}^0 \\ m\mathbf{1}_m \end{bmatrix}$, we have

$$\tilde{A}(\tilde{\mathbf{s}}^0)^{-1}\mathbf{1}_{2m} = \begin{bmatrix} A & -\underline{\mathbf{b}}\mathbf{1}_m^T \end{bmatrix} \begin{bmatrix} (S^0)^{-1}\mathbf{1}_m \\ m^{-1}\mathbf{1}_m \end{bmatrix} = A(S^0)^{-1}\mathbf{1}_m - \underline{\mathbf{b}} \cdot \frac{\mathbf{1}_m^T \mathbf{1}_m}{m} = \underline{\mathbf{b}} - \underline{\mathbf{b}} = 0$$

Thus, $\underline{\eta}(\mathbf{s}^0, m) = \left\| \tilde{A}(\tilde{\mathbf{s}}^0)^{-1}\mathbf{1}_{2m} \right\|_{(\tilde{A}(\tilde{\mathbf{s}}^0)^{-2}\tilde{A}^T)^{-1}} = 0$ \square

We present the **FindCentralPath** algorithm in Figure 5. Starting with $\underline{z} = z^0$, we take steps along the $\underline{\mathbf{b}}$ central path, decreasing z until it is sufficiently small that the analytic center of $\Omega_{\underline{\mathbf{b}}, z}^D$ is close to the analytic center of Ω^D , and therefore also close to the analytic center of $\Omega_{\underline{\mathbf{b}}, z}^D$ for some sufficiently small z .

Let us show that the **Unshift** procedure indeed takes steps near the $\underline{\mathbf{b}}$ central path:

Lemma C.14 (compare Lemmas C.8). *Given $(\mathbf{y}, \mathbf{s}, \underline{s}_{gap}) \in \mathring{\Omega}_{\underline{\mathbf{b}}, z}^D$ satisfying $\underline{\eta}(\mathbf{s}, \underline{s}_{gap}) \leq \frac{1}{40}$. Let $\underline{s}'_{gap} = \underline{\mathbf{b}}^T \mathbf{y} - z^+$ and $(\mathbf{y}^+, \mathbf{s}^+, \underline{s}'_{gap}, z^+) = \text{Unshift}(\mathbf{y}, \mathbf{s}, \underline{s}_{gap}, z)$. Then*

$$(i) \quad \underline{\eta}(\mathbf{s}, \underline{s}'_{gap}) \leq \frac{51}{400}$$

$$(ii) \quad \underline{\eta}(\mathbf{s}^+, \underline{s}'_{gap}) \leq \frac{1}{40}.$$

Proof of C.14(i). Following the proof of Lemma C.8(i) through equation 9, we have

$$\underline{\eta}(\mathbf{s}, \underline{s}'_{gap}) \leq \frac{11}{10} \cdot \underline{\eta}(\mathbf{s}, \underline{s}_{gap}) + \frac{1}{10} \leq \frac{11}{10} \cdot \frac{1}{40} + \frac{1}{10} = \frac{51}{400}$$

\square

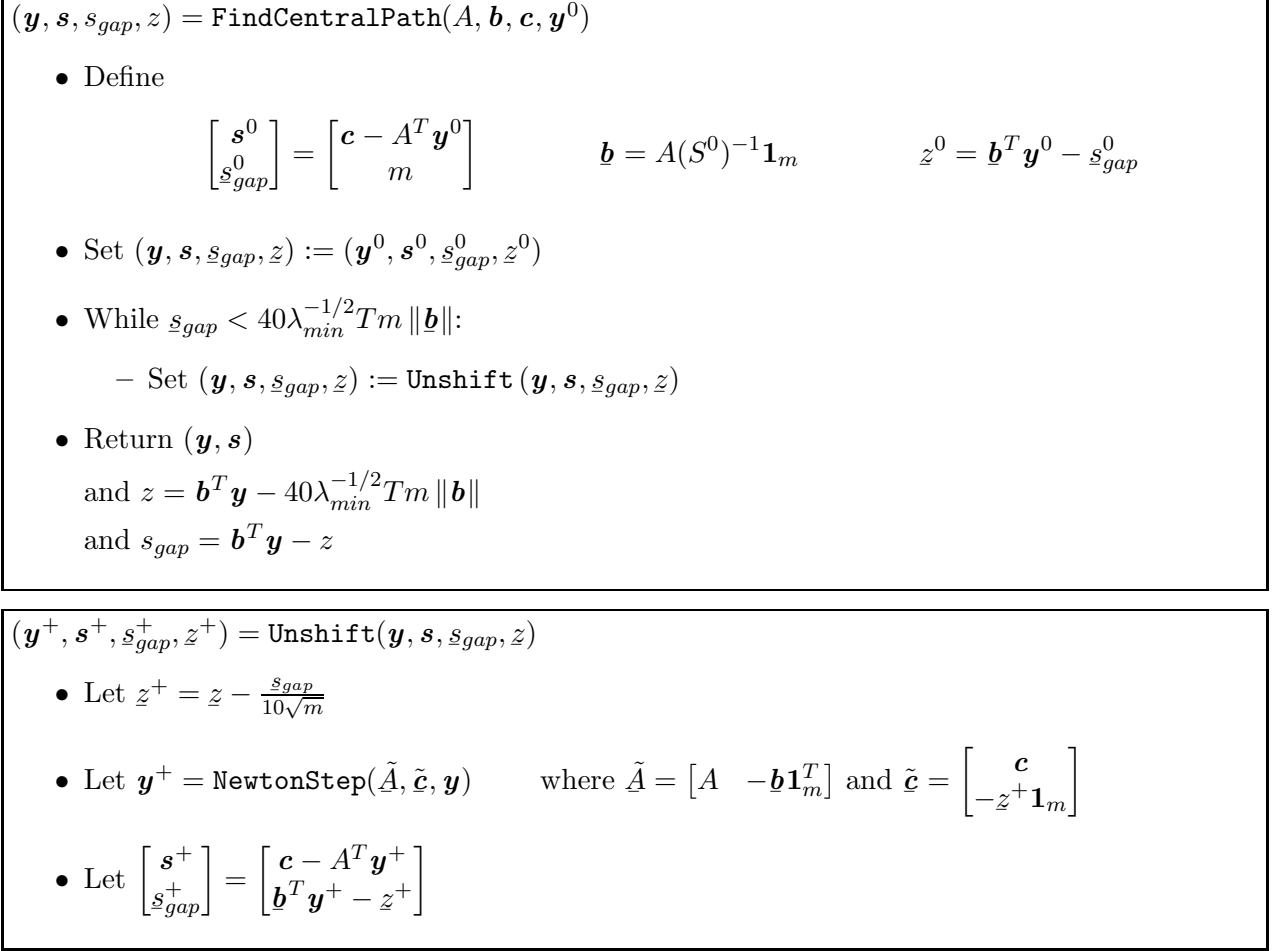


Figure 5: Algorithm for finding point near central path given feasible interior point

Proof of C.14(ii). By Lemma C.5, we have

$$\tilde{\eta}(\mathbf{s}^+, s_{gap}^+) \leq \tilde{\eta}(\mathbf{s}, s'_{gap})^2 + \frac{1}{20} \tilde{\eta}(\mathbf{s}, s'_{gap}) \leq \left(\frac{51}{400} \right)^2 + \frac{1}{20} \cdot \frac{51}{400} < \frac{1}{40}$$

□

Next, let us prove that the point returned by `FindCentralPath` is indeed near the original central path (i.e. the path given by \mathbf{b}):

Lemma C.15. *For \mathbf{y}^0 satisfying $A^T \mathbf{y}^0 < \mathbf{c}$, let $(\mathbf{y}, \mathbf{s}, s_{gap}, z) = \text{FindCentralPath}(A, \mathbf{b}, \mathbf{c}, \mathbf{y}^0)$. Then $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \mathring{\Omega}_{\mathbf{b}, z}^D$ and $\tilde{\eta}(\mathbf{s}, s_{gap}) \leq \frac{1}{10}$.*

Proof. Using the values at the end of the algorithm, we write $\tilde{\mathbf{s}} = \begin{bmatrix} \mathbf{s} \\ s_{gap} \mathbf{1}_m \end{bmatrix}$ and $\tilde{\mathbf{z}} = \begin{bmatrix} \mathbf{s} \\ s_{gap} \mathbf{1}_m \end{bmatrix}$.

To begin, we note

$$\begin{aligned}
\underline{s}_{gap} &\geq 40\lambda_{min}^{-1/2}Tm \|\underline{\mathbf{b}}\| \\
&= 40m(T^{-2}\lambda_{min})^{-1/2} \|\underline{\mathbf{b}}\| \\
&\geq 40m \|\underline{\mathbf{b}}\|_{(AS^{-2}A^T)^{-1}}
\end{aligned} \tag{15}$$

where the last inequality follows because the smallest eigenvalue of $AS^{-2}A^T$ is at least $T^{-2}\lambda_{min}$.

Similarly,

$$s_{gap} = 40\lambda_{min}^{-1/2}Tm \|\mathbf{b}\| \geq 40m \|\mathbf{b}\|_{(AS^{-2}A^T)^{-1}} \tag{16}$$

We have

$$\begin{aligned}
\tilde{\eta}(\mathbf{s}, s_{gap}) &= \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} \\
&\leq \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{(AS^{-2}A^T)^{-1}} \\
&\quad (\text{because } \tilde{A}\tilde{S}^{-2}\tilde{A}^T - AS^{-2}A^T = ms_{gap}^{-2}\mathbf{b}\mathbf{b}^T \text{ is positive semidefinite}) \\
&= \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} - ms_{gap}^{-1}\mathbf{b} + ms_{gap}^{-1}\underline{\mathbf{b}} \right\|_{(AS^{-2}A^T)^{-1}} \\
&\leq \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{(AS^{-2}A^T)^{-1}} + ms_{gap}^{-1} \|\mathbf{b}\|_{(AS^{-2}A^T)^{-1}} + ms_{gap}^{-1} \|\underline{\mathbf{b}}\|_{(AS^{-2}A^T)^{-1}} \\
&\leq \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{(AS^{-2}A^T)^{-1}} + \frac{1}{40} + \frac{1}{40} \quad (\text{by equations 15 and 16}) \\
&\leq \left(1 + ms_{gap}^{-1} \|\underline{\mathbf{b}}\|_{(AS^{-2}A^T)^{-1}} \right)^{1/2} \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} + \frac{1}{40} + \frac{1}{40} \\
&\quad (\text{by Lemma B.3, using the fact that } \tilde{A}\tilde{S}^{-2}\tilde{A}^T - AS^{-2}A^T = ms_{gap}^{-2}\mathbf{b}\mathbf{b}^T) \\
&\leq \left(1 + \frac{1}{40} \right)^{1/2} \left\| \tilde{A}\tilde{S}^{-1}\mathbf{1}_{2m} \right\|_{(\tilde{A}\tilde{S}^{-2}\tilde{A}^T)^{-1}} + \frac{1}{40} + \frac{1}{40} \quad (\text{by equation 15}) \\
&= \left(1 + \frac{1}{40} \right)^{1/2} \cdot \underline{\eta}(\mathbf{s}, \underline{s}_{gap}) + \frac{1}{40} + \frac{1}{40} \\
&\leq 2 \cdot \underline{\eta}(\tilde{\mathbf{s}}, \underline{s}_{gap}) + \frac{1}{20} \\
&\leq 2 \cdot \frac{1}{40} + \frac{1}{20} \quad (\text{by Lemma C.14(ii)}) \\
&= \frac{1}{10}
\end{aligned}$$

□

To measure the progress of the FindCentralPath algorithm, we define $B(\underline{z})$:

$$\underline{B}(\underline{z}) = \sum_{j=1}^m \log \tilde{\mathbf{s}}_j + m \log \tilde{s}_{gap} \quad \text{where } (\tilde{\mathbf{y}}, \tilde{\mathbf{s}}, \tilde{s}_{gap}) \text{ is the analytic center of } \Omega_{\underline{\mathbf{b}}, \underline{z}}^D$$

Lemma C.16 (compare Lemma C.10). *Given $(\mathbf{y}, \mathbf{s}, \underline{s}_{gap}) \in \mathring{\Omega}_{\underline{\mathbf{b}}, \underline{z}}^D$ satisfying $\underline{\eta}(\mathbf{s}, \underline{s}_{gap}) \leq \frac{1}{40}$. Let $(\mathbf{y}^+, \mathbf{s}^+, \underline{s}_{gap}^+, \underline{z}^+) = \text{Unshift}(\mathbf{y}, \mathbf{s}, \underline{s}_{gap}, \underline{z})$. Then $\underline{B}(\underline{z}^+) \geq \underline{B}(\underline{z}) + \Theta(\sqrt{m})$.*

Proof. We will follow the proof of Lemma C.10, with some minor changes.

Before we proceed, let us recall the definition $\underline{z}^+ = \underline{z} - \frac{s_{gap}}{10\sqrt{m}}$ to note that

$$s'_{gap} = s_{gap} + \underline{z} - \underline{z}^+ = 10\sqrt{m}(\underline{z} - \underline{z}^+) + \underline{z} - \underline{z}^+ \leq 11\sqrt{m}(\underline{z} - \underline{z}^+) \quad (17)$$

Now, we switch the places of \underline{z} and \underline{z}^+ , and follow the proof of Lemma C.10 up to Equation 14:

$$e^{\frac{B(\underline{z}) - B(\underline{z}^+)}{2m}} \leq 1 + \frac{1}{2s_{gap}^*} \cdot (\underline{z}^+ - \underline{z})$$

We continue:

$$\begin{aligned} &\leq 1 - \frac{s'_{gap}}{22\sqrt{m} \cdot s_{gap}^*} \quad (\text{by Equation 17}) \\ &\leq 1 - \frac{1}{22\sqrt{m}} \cdot \frac{349}{400} \quad (\text{by Lemmas C.14 and C.4}) \\ &\leq e^{-\frac{169}{4400\sqrt{m}}} \end{aligned}$$

□

Corollary C.17. *The FindCentralPath algorithm makes $O\left(\sqrt{m} \log \frac{TUm}{\lambda_{min}s_{min}^0}\right)$ calls to Unshift, where s_{min}^0 is the smallest entry of $\mathbf{s}^0 = \mathbf{c} - A^T \mathbf{y}^0$.*

Proof. Recall that the algorithm will terminate only when the value of s_{gap} has increased from its initial value of m to at least $40\lambda_{min}^{-1/2}Tm\|\underline{\mathbf{b}}\|$. So, by Lemma C.11, this will have happened once $B(\underline{z})$ has increased by $\Omega\left(m \log\left(\lambda_{min}^{-1/2}T\|\underline{\mathbf{b}}\|\right)\right)$.

According to Lemma C.16, this occurs after $O\left(\sqrt{m} \log\left(\lambda_{min}^{-1/2}T\|\underline{\mathbf{b}}\|\right)\right)$ iterations.

To complete the proof, we note that

$$\|\underline{\mathbf{b}}\| = \|A(S^0)^{-1}\mathbf{1}_m\| \leq \frac{n^{1/2}mU}{s_{min}^0}$$

□

C.4 Calls to the Solver

In each call to Unshift, we solve one system in a matrix of the form

$$\tilde{A}\tilde{S}^{-2}\tilde{A}^T = AS^{-2}A^T + ms_{gap}^{-2}\underline{\mathbf{b}}\underline{\mathbf{b}}^T$$

and in each call to Shift, we solve one system in a matrix of the form

$$\tilde{A}\tilde{S}^{-2}\tilde{A}^T = AS^{-2}A^T + ms_{gap}^{-2}\underline{\mathbf{b}}\underline{\mathbf{b}}^T$$

At the end of the interior-point algorithm we have one final call of the latter form.

In order to say something about the condition number of the above matrices, we must bound the slack vector \mathbf{s} . We are given an upper bound of T on the elements of \mathbf{s} , so it remains to prove a lower bound:

Lemma C.18. *Throughout the InteriorPoint algorithm, $\mathbf{s} \geq \frac{\epsilon}{48nmTU}\mathbf{s}^0$*

Proof. At all times during the algorithm, we know from Lemma C.4 that the elements of \mathbf{s} are bounded by a constant factor from the slacks at the current central path point \mathbf{s}^* . In particular, taking into account Lemmas C.8 and C.14, we surely have $\mathbf{s} \geq \frac{1}{2}\mathbf{s}^*$. So let us bound from below the elements of \mathbf{s}^* .

During the `FindCentralPath` subroutine, as we decrease z and expand the polytope $\Omega_{\mathbf{b},z}^D$, clearly the initial point \mathbf{s}^0 remains in the interior of $\Omega_{\mathbf{b},z}^D$ throughout. Thus, by Lemma C.3, we have $\mathbf{s}^* \geq \frac{1}{2m}\mathbf{s}^0$, and so $\mathbf{s} \geq \frac{1}{2}\mathbf{s}^* \geq \frac{1}{4m}\mathbf{s}^0$.

Unfortunately, during the main part of the algorithm, as we increase z and shrink the polytope $\Omega_{\mathbf{b},z}^D$, the initial point may not remain inside the polytope. In particular, once we have $z \geq \mathbf{b}^T \mathbf{y}^0$, the initial point is no longer in $\Omega_{\mathbf{b},z}^D$, but we may define a related point $(\mathbf{y}^z, \mathbf{s}^z, s_{gap}^z)$ that is in $\Omega_{\mathbf{b},z}^D$.

Given our current point $(\mathbf{y}, \mathbf{s}, s_{gap}) \in \Omega_{\mathbf{b},z}^D$ for $z \geq \mathbf{b}^T \mathbf{y}^0$, let us define $r = \frac{\mathbf{b}^T \mathbf{y} - z}{2(\mathbf{b}^T \mathbf{y} - \mathbf{b}^T \mathbf{y}^0)}$ and note that $0 < r < \frac{1}{2}$. We then define

$$\mathbf{y}^z = r\mathbf{y}^0 + (1-r)\mathbf{y} \quad \begin{bmatrix} \mathbf{s}^z \\ s_{gap}^z \end{bmatrix} = \begin{bmatrix} \mathbf{c} - A^T \mathbf{y}^z \\ \mathbf{b}^T \mathbf{y}^z - z \end{bmatrix} = \begin{bmatrix} r\mathbf{s}^0 + (1-r)\mathbf{s} \\ \frac{1}{2}(\mathbf{b}^T \mathbf{y} - z) \end{bmatrix} > 0$$

Therefore Lemma C.3 gives

$$\mathbf{s}^* \geq \frac{1}{2m}\mathbf{s}^z = \frac{r\mathbf{s}^0 + (1-r)\mathbf{s}}{2m} \geq \frac{r}{2m}\mathbf{s}^0$$

We then find

$$r = \frac{s_{gap}}{2(\mathbf{b}^T \mathbf{y} - \mathbf{b}^T \mathbf{y}^0)} \geq \frac{s_{gap}}{4nTU} \geq \frac{\epsilon}{24nTU}$$

The last inequality follows because, when s_{gap} decreased below $\frac{\epsilon}{3}$ on the final step, using Lemma C.4 we find that it certainly could not have decreased by more than a factor of $\frac{1}{2}$.

We conclude $\mathbf{s} \geq \frac{1}{2}\mathbf{s}^* \geq \frac{r}{4m}\mathbf{s}^0 \geq \frac{\epsilon}{48nmTU}\mathbf{s}^0$ □

We may now summarize the calls to the solver as follows:

Theorem C.19. *The `InteriorPoint`($A, \mathbf{b}, \mathbf{c}, \mathbf{y}^0, \epsilon$) algorithm makes $\mathcal{O}\left(\sqrt{m} \log \frac{TUm}{\lambda_{\min} s_{\min}^0 \epsilon}\right)$ calls to the approximate solver, of the form*

$$\text{Solve}\left(AS^{-2}A^T + \mathbf{v}\mathbf{v}^T, \cdot, \Theta(m^{-1/2})\right)$$

and one call of the form

$$\text{Solve}\left(AS^{-2}A^T + \mathbf{v}\mathbf{v}^T, \cdot, \Omega\left(\frac{s_{\min}^0 \epsilon}{m^{1/2} n^2 T^2 U^2}\right)\right)$$

where S is a positive diagonal matrix with condition number $\mathcal{O}\left(\frac{T^2 U m^2}{\epsilon}\right)$, and \mathbf{v} satisfies

$$\|\mathbf{v}\| = \mathcal{O}\left(\frac{U(mn)^{1/2}}{s_{\min}^0 \epsilon}\right)$$

Proof. From Lemmas C.17 and C.12, the total number of solves is $\mathcal{O}\left(\sqrt{m} \left(\log \frac{TUm}{\lambda_{\min} s_{\min}^0} + \log \frac{s_{gap}^C}{\epsilon}\right)\right)$,

where we know from the `FindCentralPath` algorithm that $s_{gap}^C = 40 \frac{Tm \|\mathbf{b}\|}{\lambda_{\min}^{1/2}} = \mathcal{O}\left(\frac{TUm n^{1/2}}{\lambda_{\min}^{1/2}}\right)$

As we noted above, all solves are in matrices that take the form $AS^{-2}A^T + \mathbf{v}\mathbf{v}^T$, where

$$\mathbf{v} = m^{1/2}s_{gap}^{-1}\mathbf{b} \quad \text{or} \quad \mathbf{v} = m^{1/2}\underline{s}_{gap}^{-1}A(S^0)^{-1}\mathbf{1}_m$$

We know that $s_{gap} = \Omega(\epsilon)$ and $\underline{s}_{gap} = \Omega(m)$, so we obtain the respective bounds

$$\|\mathbf{v}\| = \mathcal{O}\left(\frac{U(mn)^{1/2}}{\epsilon}\right) \quad \|\mathbf{v}\| = \mathcal{O}\left(\frac{U(mn)^{1/2}}{s_{min}^0}\right)$$

The condition number of S comes from Lemma C.18 and the upper bound of T on the slacks.

The error parameter for the solver is $\Theta(m^{-1/2})$ from the all **NewtonStep** calls. In the final solve, the error parameter is $\frac{s_{min}m^{1/2}}{TUn} \geq \frac{m^{1/2}}{TUn} \cdot \frac{s_{min}^0\epsilon}{48nmTU}$, again invoking Lemma C.18. \square